UNIVERSITÉ DE MONTRÉAL

ANALYSE ET PERTURBATION D'UN ÉCOSYSTÈME DE FRAUDE AU CLIC

MATTHIEU FAOU
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
AOÛT 2016

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

ANALYSE ET PERTURBATION D'UN ÉCOSYSTÈME DE FRAUDE AU CLIC

présenté par : FAOU Matthieu
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. MERLO Ettore, Ph. D., président
M. FERNANDEZ José M., Ph. D., membre et directeur de recherche
M. LEMAY Antoine, Ph. D., membre et codirecteur de recherche
M. CALVET Joan, Ph. D., membre et codirecteur de recherche
M. PAL Christopher J., Ph. D., membre

REMERCIEMENTS

Je voudrais tout d'abord remercier mon directeur de recherche, José Fernandez, ainsi que mes co-directeurs, Antoine Lemay et Joan Calvet, pour m'avoir donné l'opportunité de travailler sur un sujet passionnant et dans d'excellentes conditions. Je remercie aussi David Décary-Hétu et Benoit Dupont, professeurs au département de criminologie à l'université de Montréal, pour leur collaboration lors de ce projet.

Mes remerciements vont aussi l'équipe d'ESET Montréal pour leur support tant technique que financier durant cette recherche.

Je voudrais aussi remercier tous mes collègues du laboratoire de recherche SecSi pour leur bonne humeur et leurs apports à ma recherche. Je tiens particulièrement à remercier François Labrèche qui m'a aidé tout au long de ma maîtrise.

Enfin, je tiens à remercier mes parents pour leur soutien tout au long de mes études.

RÉSUMÉ

La publicité en ligne est devenue une ressource économique importante et indispensable pour de nombreux services en ligne. Cependant, on note que ce marché est particulièrement touché par la fraude et notamment la fraude au clic. Ainsi, en 2015, il est estimé que, dans le monde, les annonceurs allaient perdre plus de sept milliards de dollars américains en raison de la fraude publicitaire.

Les méthodes de luttes actuelles contre la fraude publicitaire sont concentrées sur la détection de logiciels malveillant et le démantèlement des réseaux de machines zombies qui y sont associés. Bien qu'indispensables pour limiter le nombre d'infections, ces démantèlements ne diminuent pas l'attrait pour cette fraude. Il est donc indispensable de s'attaquer en plus à l'incitatif économique. Pour cela, nous avons d'une part essayé de mieux comprendre l'écosystème de la fraude au clic et d'autre part évalué des possibilités de perturbations de cet écosystème afin de diminuer l'attractivité de la fraude.

Dans un premier temps, nous avons collecté des données réseau générées par un logiciel malveillant de fraude au clic, Boaxxe. Ces données sont des chaînes de redirection HTTP qui montrent les liens entre les différents acheteurs et revendeurs d'une publicité, c'est-à-dire la chaîne de valeur. Celles-ci commencent au moteur de recherche d'entrée, opéré par des fraudeurs, passe à travers plusieurs régies publicitaires et termine sur le site d'un annonceur, celui ayant acheté le trafic.

Dans un second temps, nous avons agrégé les données collectées afin de constituer un graphe montrant les relations entre les différents noms de domaine et adresses IP. Ce graphe est ensuite consolidé, grâce à des données de source ouverte, en regroupant les nœuds réseaux appartenant au même acteur. Le graphe ainsi obtenu constitue une représentation de l'écosystème de la fraude au clic de Boaxxe.

Dans un troisième temps, nous avons évalué différentes stratégies de perturbation de l'écosystème. L'objectif de la perturbation est d'empêcher la monétisation de trafic généré par Boaxxe, c'est-à-dire d'empêcher le transit du trafic du moteur de recherche vers le site de l'annonceur. Il s'avère que la stratégie la plus adaptée à notre problème est celle utilisant la méthode du Keyplayer. Nous avons ainsi montré qu'il était possible de protéger un nombre important d'annonceurs en supprimant un faible nombre d'intermédiaires.

Enfin, nous discutons des possibilités de mise en pratique de l'opération de perturbation. Nous insistons sur le fait qu'il est important de sensibiliser les annonceurs à la fraude afin

qu'ils puissent prendre des mesures contraignantes envers les régies publicitaires les moins scrupuleuses.

ABSTRACT

Online advertising is a growing market with global revenues of 159.8 billion dollars in 2015. Thus, it is a good target for fraudsters to make money on. In 2015, it is estimated that, globally, advertisers were defrauded of more than seven billion dollars.

The security community is concerned by this kind of fraud, known as click fraud, and a lot of research aims to limit it. Current methods are more focused on studying malware binaries and performing botnet take-downs. These operations are useful to limit the propagation of malware and to protect users from known threats. However, it does not have an impact on the economic incentives of perpetrating click fraud. In order to diminish the attractiveness of the fraud we first tried to better understand the click-fraud ecosystem and then evaluate disruption strategies on this ecosystem.

Firstly, we collected network traces generated by a well-known click-fraud malware, Boaxxe. This data are HTTP redirection chains showing the links between all the intermediaries involved in the reselling of an ad. This constitutes the value chain. The redirection chains begin at a doorway search engine, operated by fraudsters, pass through several ad networks and land on an advertiser web site, that bought the traffic.

Secondly, we aggregated the data collected into a single graph. It shows the relationships between the domain names and IP addresses involved in the Boaxxe fraud. We then consolidate this graph by merging all the network nodes operated by a single organization by leveraging information obtained from open sources. Thus, the graph is a representation of the fraud ecosystem.

Thirdly, we evaluated disruption strategies on this ecosystem. The aim is to stop the monetization of the traffic generated by Boaxxe. This is equivalent to stopping the traffic going from the doorway search engine to the web sites of the advertisers. Among the strategies tested, the most suitable for our problem was the Keyplayer strategy. We showed that it is possible to protect numerous advertisers from this fraud by disrupting the ecosystem graph.

Finally, we discuss how to perform the disruption operation in practice. We focus on increasing the level of awareness of advertisers that could have a strong position to limit click fraud. One way in which they could do so is by implementing controls to make sure they are not maintaining business relationships with unscrupulous ad networks.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

APT	Advanced Persistent Threat
API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
CERN	Organisation européenne pour la recherche nucléaire
OSI	Open Systems Interconnection
IP	Internet Protocol
URL	Uniform Resource Locator
HTML	Hypertext Markup Language
DNS	Domain Name System
CPM	Coût Pour Mille
CPC	Coût Par Clic
CPA	Coût Par Action
PCAP	Packet CAPture
SSL	Secure Sockets Layer
SEO	Search Engine Optimization

CHAPITRE 1 INTRODUCTION

Ces dernières années, on constate que l'utilisation d'internet et des services en ligne a fortement évolué avec une augmentation du nombre d'internautes de 209% entre 2005 et 2015 (Statistica, 2016). Ainsi, de nombreux services autrefois payants tels que les petites annonces, les journaux ou encore les émissions audio-visuelles sont devenus en partie accessibles gratuitement. Néanmoins, les créateurs de contenus doivent donc trouver des sources de financement pour se rémunérer ou pour payer l'infrastructure supportant le service en ligne. Parmi les différentes stratégies de financement, on compte des modèles ayant recours à la publicité en ligne tandis que d'autres reposent sur le paiement d'un droit d'accès au contenu. Ainsi, le financement de nombreux sites internet, ceux reposant sur la publicité, dépend de leurs revenus publicitaires. Il est donc primordial de veiller à l'intégrité et à la qualité du marché de la publicité en ligne.

1.1 Définitions et concepts de base : la publicité en ligne

La publicité en ligne est un marché en pleine expansion, estimé à 59,6 milliards de dollars américains en 2015 pour les États-Unis et ayant eu une croissance de 20,4% entre 2014 et 2015 (Silverman, 2016).

À l'instar de la publicité dans les médias traditionnels, différents acteurs interviennent lors d'une campagne publicitaire. Les principaux sont l'annonceur, l'éditeur et la régie publicitaire.

Annonceur. C'est une personne ou une entreprise qui cherche à promouvoir ses produits ou ses services. Pour cela, elle achète des encarts publicitaires afin d'augmenter sa notoriété ou d'augmenter le nombre de visiteurs de son site internet.

Éditeur. C'est une personne ou une entreprise qui produit du contenu via, par exemple, un site internet ou une application mobile. Elle peut être rémunérée en affichant des encarts publicitaires sur son média.

Régie publicitaire. C'est une entreprise qui fait le lien entre les annonceurs et les éditeurs. Son travail est de délivrer la bonne publicité au bon visiteur.

Différents formats de publicité existent dont les bannières publicitaires, comme présenté à la Figure 1.1, et les liens promotionnels, comme présenté à la 1.2. Les bannières publicitaires sont des encarts placés sur des pages web pouvant contenir du texte, des images ou encore une vidéo. Les liens promotionnels sont quant à eux situés sur la page de résultats des moteurs de recherche et sont liés aux mots clés de la recherche qui a été effectuée.

Figure 1.1 Bannière publicitaire ciblée

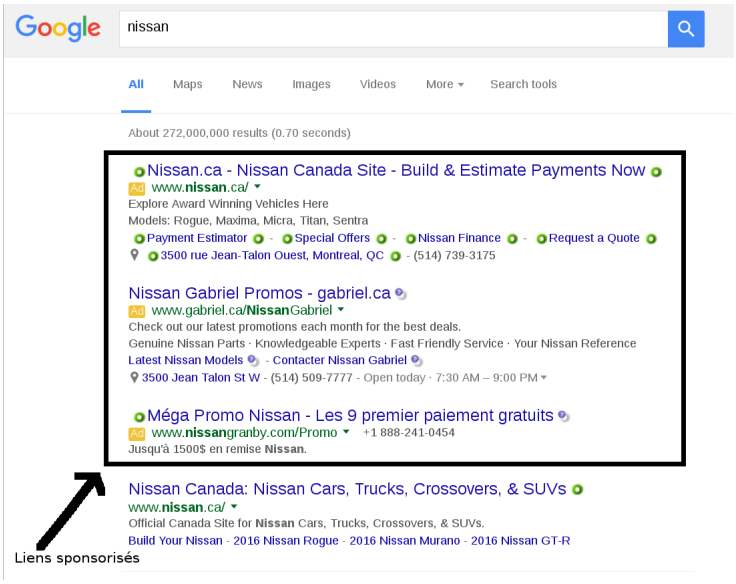


Figure 1.2 Liens promotionnels

De plus, la publicité en ligne a un caractère hautement dynamique. En effet, la publicité peut être ciblée en fonction de données caractérisant le visiteur (Support Google, 2016). Ainsi, un utilisateur faisant régulièrement des recherches pour des chaussures verra de la publicité pour des chaussures dans ses futures navigations. On notera que dans la Figure 1.1, la publicité a été ciblée en fonction du pays de l'utilisateur. En effet, l'annonce correspond à une banque canadienne et a été affichée sur un site web français visité à partir d'une adresse *Internet Protocol* (IP) québécoise.

Enfin, les encarts publicitaires peuvent être achetés et revendus en temps réel via des marchés automatisées. Ceux-ci fonctionnent selon des enchères dans lesquelles un encart publicitaire est mis en vente par un éditeur, ou une régie publicitaire, et est vendu au plus offrant, que ce soit un annonceur ou une autre régie publicitaire. Ce processus est ainsi répété pour l'affichage de chaque encart publicitaire de chaque visiteur. On notera que les encarts publicitaires peuvent être achetés et revendus plusieurs fois, en temps réel, sur ces places de vente automatisées. Ainsi, les régies publicitaires peuvent faire des profits en faisant simplement de l'achat et de la revente. Un résumé du fonctionnement du marché de la publicité est disponible à la Figure 1.3.

1.2 Problématique

Conjointement au fait que la publicité en ligne soit devenue une source de revenu importante pour les éditeurs de contenu, elle est aussi devenue une manière de monétiser les logiciels malveillants. Selon une étude de l'entreprise anti-fraude publicitaire White-Ops, la fraude publicitaire en ligne devrait coûter plus de 7 milliards de dollars américains par an (ANA - White OPS, 2015). En comparaison, le coût lié aux fraudes de cartes de crédit, une menace bien plus redoutée par le grand public, est estimé à 16 milliards de dollars américains par an (The Nilson Report, 2015).

Ainsi, la fraude publicitaire a une grande importance puisqu'elle permet à des groupes criminels de se financer. De plus, elle met le doute, aux yeux des annonceurs, sur la qualité des campagnes publicitaires en ligne. En remettant en cause la publicité en ligne, elle fait vaciller l'équilibre financier, déjà précaire pour certains, de nombreux sites internet. On notera que l'utilisation de bloqueurs de publicité a aussi un impact sur les revenus des éditeurs. Toutefois, ce phénomène est quatre fois moins coûteux pour les annonceurs que la fraude publicitaire (Baysinger, 2015).

D'un point de vue technique, la fraude publicitaire peut être réalisée de plusieurs manières. Néanmoins, le but est de faire payer un annonceur pour un clic ou une vue qui n'a pas été

Figure 1.3 Fonctionnement du marché de la publicité en ligne.

explicitement initié par un utilisateur. Par exemple, cela peut se faire grâce à un logiciel malveillant qui lance un navigateur en arrière-plan afin de cliquer automatiquement sur des encarts ou des liens publicitaires. Une autre possibilité est de remplacer ou d'ajouter des publicités sur un site internet appartenant à une autre entité, ce qu'on appelle de l'injection publicitaire. De plus amples détails concernant la fraude publicitaire sont disponibles à la section 2.2.

1.2.1 Méthodes de lutte contre la fraude publicitaire

Depuis quelques années, des chercheurs essayent de lutter contre ce phénomène via deux voies principales : la détection de clic frauduleux et le démantèlement de réseaux de machines zombies (*botnets*).

La détection de clic ou vue frauduleuse consiste à créer des filtres qui arriveront à détecter si un clic a été initié par un humain ou par un programme informatique. Par exemple, ces filtres peuvent prendre en compte l'adresse IP de la machine ou encore les mouvements de souris. En effet, les humains déplacent généralement beaucoup plus leur souris que les robots peu sophistiqués et la déplacent vers des zones précises de la page, comme les images (Zhang et al., 2011).

Le démantèlement de réseaux de machines zombies consiste à rendre inactifs les logiciels malveillants en combinant une détection anti-virus poussée avec une attaque des infrastructures de contrôle permettant à un opérateur de contrôler l'activité des machines infectées.

Cependant, certains logiciels malveillants utilisent des réseaux de communication sophistiqués, comme des réseaux pair-à-pair (Davis et al., 2008a), diminuant potentiellement ainsi l'impact de la suppression d'un nœud réseau de commandement et contrôle. De plus, même si le protocole de communication utilisé est peu résilient, il suffit à l'opérateur de déployer de nouveaux serveurs de commandement et contrôle, ce qui a un coût relativement faible, et d'infecter de nouveaux utilisateurs avec un logiciel malveillant mis à jour.

1.2.2 Limitations des méthodes de lutte

Entre les prédictions pour 2015 et pour 2016 de l'entreprise anti-fraude White Ops, on constate une augmentation du coût de la fraude pour les annonceurs de 1 milliard de dollars américains (ANA - White OPS, 2014, 2015). Il semble donc que les méthodes de lutte évoquées précédemment n'aient pas permis de freiner ce phénomène.

En effet, un rapport de Hewlett Packard Entreprise (Hewlett Packard Enterprise, 2016) a montré que la fraude publicitaire avait un rapport bénéfice sur risque très important. De plus, c'est la seule des fraudes étudiées dans ce rapport à avoir un gros potentiel de gain et un très faible risque. Ainsi, elle risque de perdurer en attirant de nombreux cybercriminels. On peut donc projeter que l'argent issu de cette fraude permettra la conception de nouveaux logiciels malveillants. Ceux-ci pourraient imiter au mieux un utilisateur classique et donc passer outre les filtres anti-fraude des régies publicitaires. On notera qu'un précédent, la lutte contre le pourriel, a aussi dû lutter contre des fraudeurs toujours plus innovants et pouvant investir de fortes sommes dans les recherche ce qui a conduit à une course à l'armement entre les spammeurs et les développeurs de filtres (Guerra et al., 2010). Ainsi il est nécessaire de lutter rapidement et surtout efficacement contre ce phénomène.

1.2.3 Nouvelles méthodes de lutte anti-fraude

Si lutter contre les logiciels malveillants et améliorer la détection des clics et vues frauduleux ne semble pas être la solution unique, il pourrait être intéressant de mieux connaître l'écosystème qui permet cette fraude. Par exemple, afin de lutter contre les fausses pharmacies sur internet, il a été proposé de s'attaquer aux entreprises qui traitent leurs transactions de carte de crédit (Krebs, 2012). Ainsi, cela a permis de couper le flot d'argent vers ces fraudeurs.

Par analogie, nous pensons qu'il serait intéressant d'identifier l'écosystème soutenant la fraude publicitaire afin d'augmenter la difficulté de la fraude mais aussi de diminuer sa rentabilité. Ici, du trafic est échangé contre de l'argent entre les différents acteurs et, on peut observer ce trafic passer de main en main. De plus, ce sont les régies publicitaires qui font le lien entre

les différents acteurs. Ainsi, le flot d'argent serait équivalent au trafic internet de fraude publicitaire et les services de paiement aux régies publicitaires. Dans la section 1.1, nous avons vu que la publicité pouvait être achetée et revendue plusieurs fois, par différentes régies publicitaires. Ainsi, l'écosystème soutenant la fraude publicitaire est constitué des éditeurs, annonceurs et régies publicitaires ainsi que de leurs relations commerciales, notamment d'achat et de revente.

Enfin, avoir une vue globale de l'écosystème permettra d'identifier les acteurs qui sont les plus à même de subir une pression politique ou économique afin qu'ils cessent d'être un point de passage du trafic malveillant. En plus d'avoir une représentation de l'écosystème, il sera aussi nécessaire de trouver des méthodes de ciblage des acteurs les plus importants dans cet écosystème. Ces méthodes pourront être affinées en prenant en compte des acteurs dits *intouchables*, c'est-à-dire sur lesquels il pourrait être difficile de faire pression. Par exemple, si l'entreprise est localisée dans un pays où il est difficile de faire pression en entreprenant des mesures légales ou réglementaires, il serait préférable de ne pas choisir cet acteur et d'en sélectionner un autre plus facilement influençable.

Ainsi, nous proposerons un ensemble de méthodes permettant d'identifier les nœuds les plus importants à cibler, et pouvant prendre en compte des nœuds intouchables, c'est-à-dire qui ne pourront pas être retirés du graphe. Ces méthodes sont issues de deux domaines différents : l'analyse de graphe et la criminologie. Dans le premier domaine, on retrouvera des méthodes classiques comme la coupe minimum dans un graphe. Dans le second on trouvera des méthodes visant en priorité les noeuds en termes de mesures de centralité, ainsi que des méthodes visant à augmenter la fragmentation du graphe, tel que la méthode du Keyplayer.

1.3 Objectifs de recherche

Le travail réalisé durant cette recherche vise à mieux comprendre la fraude publicitaire au clic afin d'en limiter les conséquences. L'objectif de la recherche est donc de répondre à la question suivante :

Comment identifier l'écosystème de la fraude au clic afin de permettre des opérations de perturbation ?

Dans cette étude, une opération de perturbation consiste à modifier l'écosystème frauduleux afin de limiter les profits générés par la fraude. Cela peut se faire, par exemple, en ciblant des acteurs importants qui sont nécessaires à son fonctionnement.

Dans cet objectif, nous proposons une méthodologie permettant d'identifier des écosystèmes de fraude au clic à partir de plusieurs logiciels malveillants. Cet objectif est séparé en plusieurs

étapes, représentées par les questions de recherche spécifiques suivantes :

1. Est-il possible de modéliser un écosystème de relations économiques (graphe) à partir de l'activité d'un logiciel malveillant de fraude au clic ?
2. Comment identifier les acteurs les plus importants de l'écosystème en vue d'une opération de perturbation ?

Bien que la fraude publicitaire soit composée de différents types de fraude, nous nous sommes concentrés, dans le cadre de ce travail, sur la fraude au clic puisqu'il est plus aisé de collecter les données nécessaires à notre étude.

1.4 Plan du mémoire

La suite du mémoire est divisée en sept chapitres. Le chapitre 2 présente des concepts plus avancés reliés à la publicité en ligne et la fraude publicitaire ainsi qu'une revue de la littérature à ce sujet. Le chapitre 3 présente la démarche utilisée au cours de cette recherche. Le chapitre 4 présente l'article de revue soumis et contient les résultats de la recherche, c'est-à-dire une analyse détaillée de l'écosystème et des acteurs clés de la fraude publicitaire pour une potentielle opération de perturbation. Le chapitre 5 est une discussion générale de notre travail. Finalement, le chapitre 6 est une conclusion qui évoquera les limitations et les potentiels travaux futurs de notre recherche pouvant les résoudre.

CHAPITRE 2 PUBLICITÉ ET FRAUDE PUBLICITAIRE : CONCEPTS ET TRAVAUX ANTÉRIEURS

Afin de comprendre la suite du travail, il est nécessaire d'introduire des notions plus avancées concernant la publicité en ligne et la fraude publicitaire. Tout d'abord, nous présenterons les modèles économiques de la publicité en ligne. Ensuite, nous présenterons des notions de réseau concernant le protocole HTTP. Enfin, nous présenterons les différents types de fraude publicitaire ainsi qu'une revue de littérature à ce sujet.

2.1 Concepts

Après avoir présenté dans un premier temps les modèles économiques de la publicité en ligne, nous détaillerons les concepts liés aux chaînes de redirection.

2.1.1 La publicité en ligne

Modèles économiques

Les acteurs présentés dans la section 1.1, à savoir les éditeurs, les annonceurs et les régies publicitaires, sont liés par des relations commerciales répondant à différents types de rémunération et par le phénomène d'achat et de revente.

Rémunération. Dans le marché de la publicité en ligne, trois principaux modèles de rémunération coexistent : le coût pour mille (vues), le coût par clic et le coût par action.

Coût Pour Mille (CPM). L'annonceur rémunère l'éditeur à chaque fois qu'une publicité est affichée mille fois. C'est le modèle le plus simple.

Coût Par Clic (CPC). L'annonceur rémunère l'éditeur à chaque fois qu'un visiteur clique sur la publicité. Généralement, le prix d'un clic est plus élevé que le coût d'une vue puisque cela montre l'intérêt du visiteur pour l'annonce publicitaire.

Coût Par Action (CPA). L'annonceur rémunère l'éditeur pour chaque action réalisée par le visiteur. Les actions peuvent être un achat ou encore l'inscription à un bulletin d'information. Dans ce cas, l'intérêt du visiteur est donc démontrablement très grand pour la publicité.

Achat et revente. Une spécificité du marché de la publicité en ligne est l'importance de l'achat et de la revente d'encarts publicitaires, pour un visiteur donné, en temps réel, appelé *arbitrage*. Ceci est facilité par l'interconnexion entre les régies publicitaires, réalisée au moyen des places de vente automatisée. Ainsi, il est courant qu'un encart publicitaire soit revendu plusieurs fois en temps réel.

L'échange de trafic est matérialisé par une redirection entre deux acteurs. Or, nous avons vu précédemment que l'échange de trafic donnait aussi lieu à une transaction financière dans le sens opposé. Ainsi, on peut considérer que d'un point de vue réseau, ces achats et reventes sont visibles. Lorsque l'utilisateur clique sur une publicité, il va être redirigé à travers tous les acteurs ayant acheté et revendu la publicité, jusqu'à atteindre le site de l'annonceur. La Figure 2.1 est un exemple de *chaîne de redirection* de publicité.

On observe aussi un modèle économique qui est, d'un point de vue réseau, identique à l'arbitrage : la syndication de moteur de recherche. Cela consiste, pour un moteur de recherche, possédant aussi une régie publicitaire, appelé moteur de recherche syndiqué, à mettre à disposition une *Application Programming Interface* (API) ou Interface de Programmation à d'autres moteurs de recherche, appelés les syndicateurs. Elle permet d'obtenir des résultats de recherches, contenant aussi des liens sponsorisés, en fonction de mots-clés. Ainsi, lors d'un clic sur un lien sponsorisé sur le site du syndicateur, l'utilisateur sera d'abord redirigé vers le moteur de recherche syndiqué. Cependant, lorsqu'on verra l'activité réseau correspondant à ce clic, on verra une chaîne de redirection, comme précédemment pour le cas de l'arbitrage.

Ainsi, dans tous les cas précédents, l'échange de trafic, et donc les achats et reventes, sont visibles grâce aux redirections entre les différents acteurs. Ainsi, il est possible de connaître l'ensemble des intermédiaires impliqués entre l'éditeur et l'annonceur. Ceci constitue une hypothèse de base de notre recherche et fera l'objet d'une discussion ultérieure.

2.1.2 Les chaînes de redirection

Dans la section précédente, nous avons vu le fonctionnement du marché de la publicité et en particulier l'importance de l'achat et de la revente qui se traduit d'un point de vue réseau par des redirections. Un ensemble successif de redirections est appelé une chaîne de redirection. Afin de comprendre en détail ce concept, nous allons présenter les notions utiles relatives au protocole HTTP.

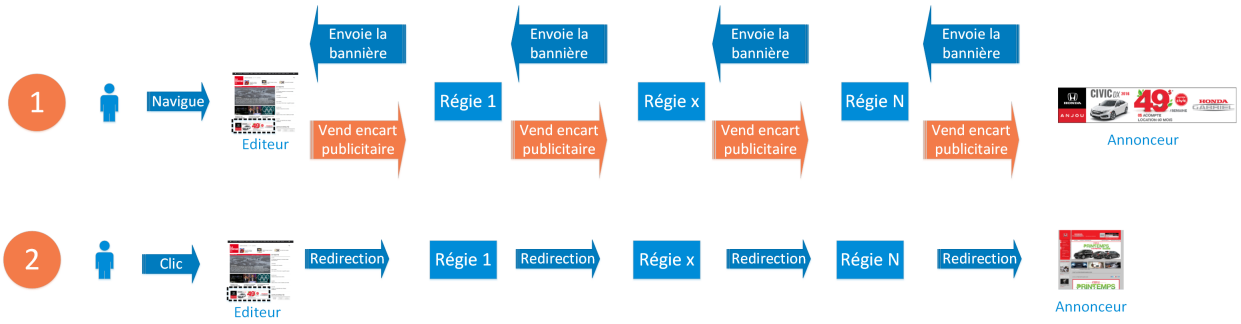


Figure 2.1 Exemple de chaîne de redirection publicitaire. La première étape est la mise en vente par l'éditeur de son encart publicitaire. Une fois l'annonceur choisi, la bannière est affichée sur le site de l'éditeur. La seconde étape intervient si l'utilisateur clique sur la publicité. Il est alors redirigé à travers tous les intermédiaires ayant acheté et revendu la publicité jusqu'à atteindre de la site de l'annonceur.

Protocole

Le protocole HTTP a été développé à l'Organisation européenne pour la recherche nucléaire (CERN) par Tim Berners-Lee et son équipe. La première version, v0.9, est datée de 1991 (Berners-Lee, 1991) mais, la version couramment utilisée a été publiée en 1997 (Fielding et al., 1997) puis mise à jour en 1999 (Fielding et al., 1999).

Le protocole HTTP est dans la couche application du modèle *Open Systems Interconnection* (OSI) et suit le modèle client-serveur. Ainsi, le client envoie une requête au serveur puis le serveur envoie la réponse au client. On notera que c'est un protocole sans état. Cela signifie que chaque couple requête/réponse est indépendant des précédents.

Les requêtes les plus courantes sont de type GET et POST qui servent respectivement à demander une ressource, telle qu'une page HTML ou une image, et à envoyer des informations à une ressource données, dans le cas d'un formulaire par exemple. En réponse, le serveur va envoyer un paquet HTTP avec un code de statut qui informe sur le résultat. Parmi les nombreuses valeurs possibles, 200 va signifier le succès, 30X (où X est un chiffre) va signifier une redirection et 404 va signifier que la ressource n'existe pas.

Une fois la réponse reçue par le client, elle va être traitée par l'agent utilisateur, un navigateur par exemple, qui affichera le contenu reçu. Au besoin, celui-ci peut aussi déclencher d'autres requêtes. Ceci permettra, entre autres, de charger les ressources externes de la page, comme des images.

Types de redirection

Dans la publicité en ligne, les redirections sont couramment utilisées afin de transférer les visiteurs au revendeur suivant. Celles-ci peuvent se présenter sous plusieurs formes :

1. Redirection HTTP
2. Redirection HTML
3. Redirection JavaScript

Des exemples des différents types de redirection sont fournis dans les Figures 2.2, 2.3, 2.4 et 2.5. La première est une redirection HTTP, la seconde une redirection HTML, la troisième une redirection JavaScript et la dernière une redirection combinant HTML et JavaScript.

Les redirections HTTP sont les plus simples puisqu'elles consistent à mettre un code de statut de la forme 30X avec un champ *Location* correspond à l'adresse web à laquelle l'utilisateur doit être redirigé. Ensuite, l'agent utilisateur va automatiquement contacter le serveur suivant afin d'obtenir la ressource demandée. On notera que par convention les redirections HTTP n'ont pas d'influence sur le champ *Referer* des requêtes HTTP, qui contient l'adresse de la page précédente. Ainsi, même après plusieurs redirections HTTP consécutives, le champ Referer contiendra l'adresse de la page initiale.

Les redirections HTML utilisent principalement les balises `meta` et `iframe`. La première permet de rediriger l'utilisateur vers une autre page tandis que la seconde permet de charger une page à l'intérieur de la page courante.

Les redirections JavaScript peuvent prendre des formes multiples. D'une part, il est possible de changer la valeur de *windows.location.href* pour la nouvelle adresse. D'autre part, il est possible de combiner des actions JavaScript et de l'insertion de HTML. Par exemple, un script peut insérer un lien hypertexte, avec la balise `a`, et cliquer automatiquement sur cette balise ce qui va conduire exactement au même comportement qu'une redirection native en JavaScript. En outre, certains des scripts sont rendus impénétrables (*obfuscated* en anglais) et il peut donc être difficile de comprendre leur but. De plus, l'adresse de redirection peut être créée dynamiquement, en ajoutant par exemple l'heure UNIX (timestamp) en paramètre. Ainsi, il n'est pas évident, lors d'une analyse statique automatisée d'un code JavaScript, de connaître l'adresse de redirection.

Arbre de navigation HTTP

Afin d'identifier les acteurs qui ont acheté et revendu une publicité à partir des traces réseaux d'une session de navigation, il peut être intéressant de représenter cette navigation sous la

```
HTTP/1.1 301 Moved Permanently
Location: HTTP://www.perdu.com/
Content-Type: text/html
Content-Length: 170

<html>
<head>
<title>Moved</title>
</head>
<body>
<h1>Moved</h1>
<p>This page has moved to
<a href="http://www.perdu.com/">http://www.perdu.com/</a>.</p>
</body>
</html>
```

Figure 2.2 Redirection HTTP

```
<html>
        <head>
                <title>Page de redirection</title>
                <META HTTP-equiv="refresh"
                        content="0;URL=http://www.perdu.com">
        </head>
</html>
```

Figure 2.3 Redirection HTML

```
var id = 123;
var url = "http://www.perdu.com/index.php?id="+id;
//http://www.perdu.com/index.php?id=123
window.location.href = url;
```

Figure 2.4 Redirection JavaScript dynamique

```
<html>
        <head>
                <title>Page de redirection</title>
        </head>
        <body>
        <div id="redir"></div>
        </body>
        <script type="javascript">
                var link = '<a href="http://www.perdu.com/index.php" '
                                +'id="lien_redirection">Redirection</a>';
                document.getElementById("redir").innerHTML(link);
                document.getElementById("lien_redirection").click();
        </script>
</html>
```

Figure 2.5 Redirection JavaScript avec insertion d'une balise HTML

forme d'un arbre dans lequel un nœud est une requête et un arc signifie que le nœud parent a déclenché le nœud enfant. Le déclenchement peut avoir lieu automatiquement ou au moyen d'une action de l'utilisateur comme un clic. Un exemple d'un tel arbre est représenté à la Figure 2.6.

Cependant, nous avons vu précédemment que le protocole HTTP est sans état. Ainsi, il n'est pas évident de relier des couples requête/réponse entre-eux. Des travaux antérieurs se sont penchés sur ce problème et différentes solutions ont été proposées.

Tout d'abord, la méthode intuitive est d'utiliser le champ Referer des requêtes HTTP afin de connaître la page précédente (Xie et al., 2013). Cependant, cette méthode masque les nœuds intermédiaires utilisant des redirections HTTP qui, comme vu précédemment, ne modifient pas le champ Referer. De plus, cette méthode est vulnérable à l'utilisation d'un agent web personnalisé qui modifierait le comportement du champ Referer, comme celui utilisé par le logiciel malveillant Bedep (Frankoff, 2015).

Ensuite, il a donc été proposé de chercher dans les réponses HTTP les *Uniform Resource Locator* (URL) possibles de redirection (Mekky et al., 2014). Lorsqu'on essaye de relier la requête à la réponse précédente, il suffit de regarder si une réponse contient l'URL de requête. Cependant, certaines URL de redirection sont générées dynamiquement, par exemple en fonction de l'heure UNIX. Ainsi, avec une analyse statique du script de redirection, il n'est pas toujours possible de trouver l'adresse exacte dans la liste des URL collectées.

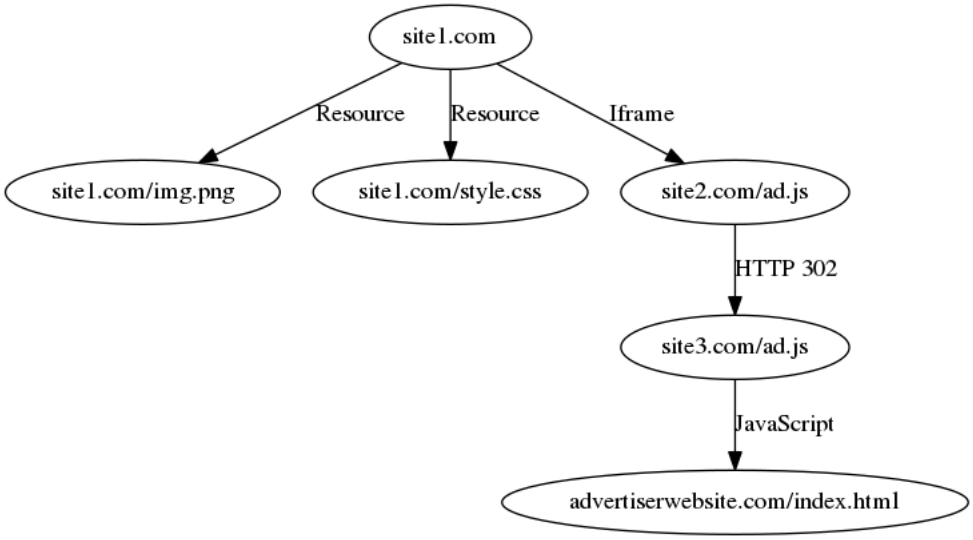


Figure 2.6 Exemple d'un arbre de navigation HTTP

Enfin, des chercheurs ont proposé de combiner une analyse statique des scripts de redirection avec une analyse dynamique opérée en rejouant le trafic réseau collecté dans un navigateur web instrumenté (Neasbitt et al., 2014). Ainsi, bien que complexe, cette technique est résistante à l'obfuscation du JavaScript et permet donc de reconstruire des arbres de navigation HTTP d'une très grande précision.

Toutefois, les techniques présentées sont pour les premières trop basiques tandis que la dernière est difficile à répliquer.

2.2 La fraude publicitaire : concepts et travaux antérieurs

Dans la sous-section précédente, nous avons présenté les concepts lié à la publicité en ligne. Ainsi, ceci nous permet d'introduire dans cette section les concepts liés à la fraude publicitaire et une revue de littérature à ce sujet. Dans un premier temps, nous donnerons des définitions, puis nous ferons une revue de littérature des travaux liés aux logiciels malveillants de fraude au clic, à la détection de clic frauduleux et à l'économie de la fraude publicitaire.

2.2.1 Définition

Étant donné l'important marché que représente la publicité en ligne, il n'est pas étonnant qu'elle soit la cible de fraudeurs. On sépare généralement la fraude publicitaire en quatre domaines : la publicité malveillante (*malvertising*), la fraude à l'impression, la fraude au clic (*click fraud*) et la fraude à l'action.

Le malvertising est une fraude opérée par l'annonceur. Cela consiste à exploiter la publicité afin, généralement, d'installer des logiciels malveillants en tirant parti de failles dans le navigateur ou dans les plug-ins tels que flash ou java. C'est un cas particulier de ce que l'on nomme *drive-by download*.

La fraude à l'impression consiste à générer des vues non désirées afin de tirer parti du modèle CPM. L'injection publicitaire est une forme de fraude à l'impression. Cela consiste à ajouter des publicités sur une page web entre le serveur (exclu) et le client (inclus) grâce à un proxy ou un logiciel malveillant par exemple.

La fraude au clic consiste à faire payer un annonceur pour un clic qui a été uniquement réalisé dans le but de générer un coût à l'annonceur et non pour intérêt pour la publicité. On notera que parfois le terme fraude au clic est utilisé abusivement pour désigner le fraude à l'impression. Si des scripts automatisés sont en grande partie responsables de cette fraude, les fermes d'humains, c'est-à-dire un groupe d'humains payés pour réaliser manuellement des clics, sont aussi considérées comme de la fraude au clic.

La fraude à l'action consiste à faire payer un annonceur pour une action, tel que l'inscription à une infolettre, qui a été réalisée uniquement dans le but de générer un coût pour l'annonceur ou pour une action qui n'a pas été causée par l'affichage de la publicité. La fraude au marketing d'affiliation est un sous-ensemble de la fraude à l'action. Dans ce modèle économique de l'affiliation, un site internet, tel qu'un site de commerce de ligne, va rémunérer les sites internets étant à l'origine du déclenchement d'un achat. Ainsi, les affiliés ont carte blanche pour inciter les acheteurs. Généralement, ce sont des blogueurs qui sont affiliés et insèrent un lien, incluant leur numéro d'affiliation, vers le produit correspondant à leur article. Étant donné que seul le dernier affilié ayant amené le visiteur vers le site du marchand est rémunéré, des fraudeurs peuvent amener les acheteurs à visiter le site de commerce contre leur volonté, au moyen d'un pop-up par exemple. Ainsi, ce sont eux qui récupèrent la prime pour l'achat. Cette fraude a déjà été étudié en détail (Snyder and Kanich, 2015) et nous la considérons donc en dehors du sujet de cette recherche.

2.2.2 Pratiques grises

La publicité en ligne est touchée par plusieurs phénomènes, qui bien que légaux, sont à la frontière avec la fraude.

Tout d'abord, plusieurs entreprises se sont spécialisées dans les logiciels publicitaires (*adware*). Ce sont des programmes que l'utilisateur installe séparément ou qui est installé en complément d'un autre logiciel et qui va afficher de la publicité sur la machine où il est présent.

Cependant, ce ne sont pas des logiciels malveillants puisqu'ils sont installés avec le consentement de l'utilisateur. On les appelle généralement des programmes potentiellement indésirables. Par ailleurs, ils peuvent tout de même nuire à l'expérience utilisateur et aux revenus publicitaires de certains sites internets. C'est pourquoi des recherches commencent à être faites sur ces programmes (Thomas et al., 2015).

Ensuite, plusieurs techniques sont utilisées afin d'améliorer le référencement de sites internets. Certains propriétaires de sites internets ont recours à des techniques dites de *Black Search Engine Optimization* (SEO) qui consistent, par exemple, à utiliser un réseau de machines zombies (Wang et al., 2013) afin d'améliorer le référencement de leur site internet sur les moteurs de recherche tels que Google. Cette amélioration de référencement permet d'augmenter le nombre de visiteurs arrivant sur le site web. Il peut aussi augmenter artificiellement le prix de vente de chaque publicité affichée sur ce site web. Ainsi, le black SEO permet d'augmenter les revenus publicitaires en trompant partiellement les annonceurs.

Enfin, il est possible de lier un nom de domaine non utilisé à un service dit de *parking*. Celui-ci va remplir le site non utilisé avec de la publicité et bénéficier des visiteurs qui y arrivent par erreur. Cependant, cette pratique peut parfois être très proche de pratiques frauduleuses (Alrwais et al., 2014).

2.2.3 La fraude au clic

Pour réaliser des profits grâce à la fraude au clic, il est nécessaire de pouvoir recevoir une partie du prix payé par l'annonceur pour le clic. Les clics doivent donc être fait sur des publicités affichées au nom du fraudeur. Pour cela, ce dernier doit s'enregistrer en tant qu'éditeur auprès d'une (ou plusieurs) régie(s) publicitaire(s) afin de pouvoir recevoir des encarts publicitaires à afficher. Ensuite, il dispose de différentes manières d'afficher cette publicité :

1. Injecter de la publicité sur des sites existants
2. Monter un site internet avec du contenu
3. Monter un moteur de recherche syndiqué

Le premier cas est à la frontière de la définition de la fraude au clic et n'est pas considérée comme de la fraude. En effet, cela consiste à remplacer les publicités existantes d'un site internet par ses propres publicités, comme dans la fraude à l'impression, et à bénéficier du revenue des clics. Pour ce faire, la page est modifiée après son envoi par le serveur au client, soit par le fournisseur d'accès à internet soit par un logiciel installé sur la machine du client. C'est ce qu'on appelle de l'injection de publicité. Alors que de plus en plus de voix s'élèvent pour imposer la neutralité aux fournisseurs d'accès à internet (White House, 2015), certains

d'entre-eux, comme AT&T, n'hésitent pas à injecter de la publicité supplémentaire dans les pages de leurs clients (McCormick, 2015). Les logiciels publicitaires ont eux aussi recours à l'injection de publicité mais, du côté du client. Cela peut être réalisé grâce à la modification des serveurs *Domain Name System* (DNS) par défaut de la machine, redirigeant ainsi les résolutions des domaines des principales régies publicitaires vers des adresses IP possédées par l'entreprise opérant le logiciel publicitaire, grâce à un proxy web ou plus simplement grâce à une extension de navigateur qui va modifier le contenu de la page à son chargement.

Les deux cas suivants sont finalement relativement proches. En effet, cela consiste à monter un site internet comme n'importe quel éditeur classique. Cependant, le deuxième cas nécessite plus d'efforts. En effet, pour maximiser les revenus ou pour éviter le bannissement pour fraude, le site doit paraître le plus légitime possible. Pour cela, le site doit comporter du contenu, relativement à jour, et cohérent avec le thème du site. À l'inverse, en raison de l'existence de la syndication, il est relativement rapide et aisé de monter un moteur de recherche. En effet, il suffit de s'enregistrer auprès d'une régie publicitaire proposant ce type de service et d'afficher dans les résultats de recherche les résultats fournis par la régie. Ainsi, d'une part, il paraîtra tout à fait légitime vis-à-vis d'une analyse peu poussée et d'autre part, on peut supposer qu'il sera possible de recréer très rapidement et pour un faible coût un moteur de recherche en cas de détection.

Enfin, il suffit maintenant de maximiser le nombre de clics sur les publicités affichées sur les sites détenus par le fraudeur. Pour cela, il est possible d'améliorer le référencement afin d'attirer des visiteurs légitimes, mais ce processus est relativement long, complexe et coûteux. L'autre solution est d'utiliser un logiciel malveillant installé sur de nombreuses machines pour cliquer sur les publicités. Pour cela, le propriétaire du "faux" moteur de recherche peut soit avoir son propre logiciel malveillant, soit payer l'opérateur d'un réseau zombie pour que ses robots cliquent sur les publicités. Étant donné que les réseaux zombies sont composés d'un nombre conséquent de machines, les clics sembleront provenir de différents utilisateurs. Ainsi, il sera plus difficile pour les filtres anti-fraude de remarquer que ces clics ont été initiés par un opérateur unique.

2.2.4 Les logiciels malveillants de fraude au clic

Étant donné l'attractivité de la fraude au clic, de nombreux logiciels malveillants ont été développés ces dernières années.

Jusqu'en 2013, ZeroAccess était un des plus gros logiciels malveillants de fraude publicitaire avec des revenus estimés à 2,7 millions de dollars américains par mois. En collectant des données avant et après son démantèlement, des chercheurs ont pu évaluer l'impact de ce

programme de fraude sur un jeu de données d'une régie publicitaire (Pearce et al., 2014). Cependant, ce démantèlement n'a pas été un succès total puisque seuls les serveurs de commandement et de contrôle ont été fermés. Or, les différents ordinateurs infectés étaient toujours reliés entre-eux grâce à un réseau pair-à-pair robuste. Ainsi, il a été constaté au début de l'année 2015 que le logiciel malveillant avait été mis à jour sur les machines infectées et qu'il avait recommencé ses activités de fraude publicitaire (Stockley, 2015).

Comme vu à la section 2.2.3, la fraude peut reposer sur plusieurs principes. Ainsi, certains logiciels malveillants reposent sur un faux moteur de recherche, à l'instar de ClickBot.A (Daswani and Stoppelman, 2007). D'autres ont recours à des techniques plus évoluées comme le changement de DNS (Alrwais et al., 2012) ou une fausse régie publicitaire (Miller et al., 2011). Contrairement aux réseaux zombies classiques de fraude publicitaire, on note l'émergence de l'échange de trafic. En générant du trafic, les utilisateurs de ces services gagnent des crédits qu'ils peuvent dépenser pour acheter du trafic pour leur propre site web (Javed et al., 2015). Cependant, ces articles se sont concentrés sur le fonctionnement technique des logiciels malveillants et ne comportent pas d'analyse détaillée de l'écosystème qui permet cette fraude.

Ainsi, le fonctionnement des logiciels malveillant de fraude au clic est bien connu. De plus, des démantèlements de réseaux zombies de fraude au clic ont aussi été effectués par le passé. Cependant, on constate que ces recherches n'ont pas permis de stopper la fraude.

2.2.5 La détection du trafic malveillant

Même si beaucoup de recherches en sécurité se sont concentrées sur les logiciels malveillants, des chercheurs ont aussi étudié la possibilité de détecter les clics ou vues frauduleuses. Ainsi, si le fraudeur n'est plus rémunéré pour sa fraude ou si cela demande trop d'investissements pour éviter les filtres, la fraude devrait grandement diminuer. Ainsi, des chercheurs ont acheté du trafic à différents fournisseurs pour des sites internets qui étaient sous leur contrôle (Zhang et al., 2011; Dave et al., 2012). Ils ont donc pu évaluer la qualité du trafic fourni par chacun afin de repérer si certaines régies publicitaires étaient moins vigilantes que d'autres. Il en ressort que la fraude semble être particulièrement importante chez un certain nombre de fournisseurs.

D'autres études ont essayé de trouver des méthodes permettant de différencier du trafic malveillant et légitime par des caractéristiques difficilement modifiables par le fraudeur. Ainsi, Dave et al. (2013) montrent que les fraudeurs ont un revenu plus élevé que les éditeurs légitimes puisqu'ils doivent couvrir le risque d'être arrêtés. Cependant, s'ils baissent leurs revenus, ils risquent de se pénaliser financièrement et la fraude serait donc moins rentable. Il

est donc possible, à partir des données d'une régie publicitaire, d'identifier les fraudeurs en surveillant leurs revenus. Cependant, cela implique que la régie publicitaire ait un intérêt à lutter contre la fraude.

Bien que les méthodes proposées semblent prometteuses, cela entraîne une course à l'armement avec les fraudeurs. Ainsi, il faudrait investir de fortes sommes d'argent en recherche sur les filtres pour espérer gagner cette course.

2.2.6 L'économie de la fraude publicitaire

Des recherches se sont attachées à modéliser l'économie de la fraude publicitaire afin de mieux en comprendre les enjeux.

Tout d'abord, Gill et al. (2013) ont étudié le marché légal de la publicité et en proposent une rétro-ingénierie afin d'en comprendre les enjeux tant économiques que sur l'exploitation des informations personnelles.

Ensuite, d'autres chercheurs ont modélisé la fraude au clic afin de savoir si certain des acteurs ont un intérêt à frauder. Les régies publicitaires sont rémunérées pour chaque clic ou vue facturé à l'annonceur. Ainsi, si elles luttent contre la fraude, elles risquent de diminuer leurs profits, au moins à court terme. Cependant, il s'avère que les régies publicitaires ont un intérêt économique à lutter contre la fraude pour garder leur part de marché (Mungamuru et al., 2008; Dritsoula and Musacchio, 2014). Cela s'explique par le fait que les annonceurs changeront de régie publicitaire si celle-ci n'est pas assez agressive contre la fraude publicitaire. Cependant, cela nécessite l'hypothèse que les annonceurs soient capables de différencier du trafic provenant d'humains du trafic provenant de robots.

Enfin, les éditeurs n'ont pas d'intérêt à frauder si leur rémunération est assez élevée (Asdemir et al., 2008). En effet, sachant qu'ils peuvent être bannis en cas de fraude, cela ne serait plus rentable pour eux de faire de la fraude.

Ainsi, ces études sont intéressantes pour comprendre l'incitatif à frauder mais, elles ne permettent pas directement de lutter contre la fraude.

2.3 Solutions

Nous avons vu que l'échange de trafic entre les différents acteurs correspondait à des transactions commerciales. De plus, ces échanges de trafic sont visibles via les chaînes de redirection. Il serait donc possible d'identifier les relations entre les différents acteurs ce qui conduirait à une représentation de l'écosystème de la fraude au clic.

Or, les travaux antérieurs se sont focalisés sur la détection de clic frauduleux et sur l'étude, d'un point de vue technique, des logiciels malveillants de fraude au clic.

Ainsi, il s'est révélé nécessaire de faire une étude approfondie de l'écosystème de la fraude publicitaire afin d'en comprendre son fonctionnement et d'en identifier ses points faibles.

CHAPITRE 3 DÉMARCHE DU TRAVAIL DE RECHERCHE

Dans les chapitres précédents, nous avons présenté la problématique de notre recherche, les notions de bases relatives à la fraude au clic ainsi qu'une revue de littérature à ce sujet. Dans ce chapitre, nous rappellerons la problématique puis nous expliquerons la démarche de notre travail en la liant à notre article de revue présenté au chapitre 4.

3.1 Problématique

Dans le chapitre 1, nous avons présenté l'objectif de notre travail qui consiste à analyser l'écosystème de la fraude au clic et à trouver des façons de le perturber afin de diminuer l'attractivité de la fraude.

Nous avons aussi établi plusieurs questions spécifiques de recherche. Notre travail consiste donc à établir une méthodologie permettant de reconstruire l'écosystème de la fraude au clic à partir de traces d'activités d'un logiciel malveillant de fraude au clic, à analyser cet écosystème et à trouver des manières de le perturber.

3.2 Démarche

Dans le chapitre précédent, nous avons montré que la littérature manquait, à ce jour, d'une analyse de l'écosystème de la fraude au clic. Il n'existe pas non plus, à ce jour, de tentatives de perturbation d'un écosystème de fraude au clic. Enfin, nous avons montré qu'il serait possible de collecter des données afin d'avoir une représentation de l'écosystème de la fraude liée à un logiciel malveillant de fraude publicitaire.

Le chapitre 4 est un article soumis à la revue Computer & Security le 27 juillet 2016. Il comprend le travail réalisé durant la maîtrise. L'article comporte six grandes sections.

La section 4.2 donne les concepts de base nécessaire à la compréhension de l'article. Certains d'entre-eux ont été présentés dans la section 2.

La section 4.3 présente la méthodologie. Notamment, il est expliqué comment ont été collectées les données, quels sont les logiciels malveillants étudiés, comment les chaînes ont été reconstruites à partir des traces réseaux et comment ces données ont été agrégées afin d'obtenir un graphe représentant l'écosystème de la fraude au clic. Cette section répond partiellement à la première question de recherche.

La section 4.4 est une analyse de l'écosystème. Elle comprend une représentation visuelle

de l'écosystème, une analyse des différents type d'acteurs et de l'évolution temporelle de la fraude. Cette section complète la réponse à la première question de recherche.

Les sections 4.5 et 4.6 présentent les stratégies de perturbation de l'écosystème. D'une part, nous y présentons une comparaison entre différentes méthodes permettant de choisir les meilleurs cibles afin de perturber au maximum l'écosystème de fraude au clic. D'autre part, nous y analysons les résultats obtenus grâce à la meilleure méthode issue de la comparaison. Cette section est la réponse à la seconde question de recherche.

La section 4.7 présente une discussion des résultats obtenus. En particulier, nous y expliquons comment il serait possible d'agir en pratique sur les acteurs sélectionnés dans la section précédente.

Ainsi, l'article a pour but de présenter d'une part la méthodologie de notre recherche et d'autre part de répondre aux deux questions de recherche établies à la section 1.3.

CHAPITRE 4 ARTICLE 1 : FOLLOW THE TRAFFIC: A COMPARATIVE STUDY OF DISRUPTION TECHNIQUES AGAINST CLICK FRAUD

Authors Matthieu Faou[1], Antoine Lemay[1], David Décary-Hétu[2], Joan Calvet[3], François Labrèche[1], José M. Fernandez[1], Benoit Dupont[2]

École Polytechnique de Montréal[1], Université de Montréal[2], ESET[3]

Montréal, Canada

{matthieu.faou, antoine.lemay, francois.labreche, jose.fernandez}@polymtl.ca[1]

{david.decary-hetu, benoit.dupont}@umontreal.ca[2]

calvet@esetlabs.com[3]

Submitted to : Computer & Security (COSE)

Abstract

Advertising fraud, particularly click fraud, is a growing concern for the online advertising industry. The use of click bots, malware that automatically clicks on ads to generate fraudulent traffic, has steadily increased over the last years. While the security industry has focused on detecting and removing malicious binaries associated with click bots, a better understanding of how fraudsters operate within the ad ecosystem is needed to be able to disrupt it efficiently.

This paper provides a detailed dissection of the advertising fraud scheme employed by Boaxxe, a malware specializing in click fraud. By monitoring its activities during a 7-month longitudinal study, we were able to create of map of the actors involved in the ecosystem enabling this fraudulent activity. We also collected data from the Ramdo click-fraud malware to validate our methodology. We then compared different metrics to select the best set of key actors of the Boaxxe ecosystem. We found that the Keyplayer technique is the most suitable to effectively influence this ecosystem in order to maximize disruption of click-fraud monetization. The results show that it would be possible to efficiently disrupt the ability of click-fraud traffic to enter the legitimate market by pressuring a limited number of these actors. We assert that this approach would produce better long term effects than the use of take-downs as it renders the ecosystem unusable for monetization.

4.1 Introduction

The development of the Internet enabled a wealth of content to become readily accessible. A large volume of this content is offered for free. This is true even for content that we used to pay for, such as newspapers. Naturally, content creators need to make up for the absence of income by finding a new revenue stream. This monetization scheme is Internet advertisement. By showing ads to their visitors, and having them click on those ads, content creators are able to convert traffic into money. This business model is now a dominant force on the Internet, with the size of the market in 2014 estimated at 59.6 billion dollars in the US alone (iab, 2016), and 159.8 billion dollars worldwide (Statista, 2016).

However, criminals can also abuse this system to monetize computers infected with malware. By distributing specialized ad fraud payloads to these machines, fraudsters can generate a large number of requests that resemble those produced by humans. They can then get revenue from this fake traffic without needing to invest in content creation. In fact, the Association of National Advertisers estimated that, in 2015, publicity fraud will cost more than 6 billion dollars to advertisers worldwide (White Ops and Association of National Advertisers), representing close to 4% of total global publicity revenue. The prevalence of this problem undermines the business model that underpins most free services on the Internet today. Thus, it is imperative to find ways to better understand and address this problem.

One technique that can be used to combat this form of fraud is the disruption of the so-called *value chain*, which shows the links between fraudulent actors and legitimate businesses through which fraudsters acquire wealth. One example of the successful use of this method was the campaign to shut down payment processors used for scareware (Krebs, 2011). In that case, the monetization scheme was the distribution of a fake anti-virus (fake AV). The user was informed that his computer was infected and was then offered a fake AV product to clean the infection. The "product" had to be purchased through a credit card transaction. The fact that these credit card transactions could be linked to particular payment processors, allowed credit card companies to stop this kind of transactions. But, in the case of publicity fraud, this is not so easy. First, money changes hands several times before reaching the fraudsters. Furthermore, there is no centralized database containing all of these transactions that could be analyzed to understand where the money going to fraudsters comes from. How then can we build a global picture of the business relationships between actors involved in ad fraud, willingly or unwillingly? How can we also identify choke points where disruptive pressure can be applied?

One possible method is to reconstruct the traffic, i.e. the redirection chains, involved in

advertising to map the corresponding value chains. By following a series of redirections taken by an automated click-fraud module from the infected computer to the advertiser, it is possible to get a glimpse of how traffic changes hands between the different actors involved in click-fraud. Every time traffic is transfered between actors, there is a corresponding economic transaction. Much like police officers doing surveillance on drug dealers, by observing enough transactions, it is, in principle, possible to reconstruct the entire network of actors involved in such illicit activities. The network can then be analyzed to find optimal targets for disruption.

In this paper, we present our efforts for doing exactly that on the click-fraud network associated with the Boaxxe malware. The paper starts by providing some background about Internet publicity and automated click-fraud in Section 4.2. We describe in Section 4.3 the data collection methods employed in our 7-month longitudinal study on Boaxxe, and our methodology for constructing the actor graph from the *redirection chains*. We also describe our data collection effort on the Ramdo malware, used in this section to validate our redirection chain reconstruction algorithm. We analyze and discuss this graph in Section 4.4. In Section 4.5, we discuss various possible strategies that have been studied for disrupting other types of criminal activity. We then study some of them and compare their relative effectiveness in Section 4.6. This allowed us to identify the *keyplayer* technique as the most effective, a result which we were able to confirm on the Ramdo data set as well. We further describe and discuss in detail the critical targets identified by our analysis. In Section 4.7, we discuss how the methods and findings presented in this paper could be applied to design and implement more effective and generic anti-click fraud policies and strategies. We conclude the paper by discussing related work in Section 4.8 and providing a summary of results, limitations and directions for future work in Section 4.9.

This paper is the combination of research contained in previous papers (Faou et al., 2016a,b) and additional research done since. It also contains details not previously described, such as details on the redirection chain reconstruction algorithm and results on the persistence of actors over time. New research includes comparison with other graph disruption techniques besides *keyplayer* and the collection and use of a second data set based on another malware (Ramdo) for the validation of the redirection chain reconstruction algorithm and results on disruption techniques.

4.2 Background

In this section, we introduce basic notions related to the legitimate Internet advertising ecosystem and we present some of the techniques used to defraud this market.

4.2.1 Online advertising

To fully understand ad fraud, it is critical to know the basics of online advertising.

Advertiser. A person or a company that wants to promote its products or services. This entity pays to *display* ads on other web sites to attract visitors to its web site. It may also pay per traffic redirected to its web site as a result of a human action, i.e. a *click* on the corresponding ad.

Publisher. A person or a company running a web site that displays advertisements to its visitors. This entity earns money by showing ads and by having users click on these ads.

Ad network. A person or a company that buys and sells ads or visitors. It buys and sells traffic in bulk through pre-established contracts, or through *ad exchanges* which are automated auction markets where traffic is bought and sold instantly. In other words, it is an intermediary between advertisers and publishers. Ideally, an ad network aims to match the right ad with the right visitor in order to serve the needs of its end client, the advertiser. This is done, for example, by matching the advertiser's requirements for visitors with the user profiles constructed from browsing information, e.g. browsing history, search history, web site cookies, etc. The matching typically occurs in real time using the ad network's market infrastructure. However, not all traffic brokers operate their own infrastructure. Some are only marketing companies and rely on Solution-as-a-Service (SaaS) providers to manage their ad serving infrastructure.

Compensation. Ad networks generally propose different types of compensation. The three main types are *Cost Per Mille*[1] (CPM), *Cost Per Click* (CPC) and *Cost Per Action* (CPA). The cheapest is CPM, where each displayed ad is typically compensated with a few tenths of a cent. This relatively low price is due in part to the fact that there is little guarantee that visitors are interested in the ad and will take further revenue-production actions for the advertiser. On the other hand, the fact that a visitor clicks on the ad (CPC) provides better chances of revenue and, accordingly, clicks are typically more expensive. Prices vary widely depending on the advertiser category, but mean prices are in the dollar range (Wordstream, 2015). Finally, clicks that can be linked to an actual revenue-generating action by the user, i.e. CPA, are compensated the most.

When an advertiser wants to attract traffic for its web site, its ad network will promote the web site by displaying *creatives*, such as text banners, pictures or ad videos, on other

1. Cost per thousand views. This terminology comes from traditional advertising.

web sites. For the advertiser, this is the equivalent of buying visitors. The ad network can also take action so that the advertiser web site appears besides the search engine results, i.e. "sponsored" sites. In other words, search engines can also be considered publishers and are compensated for bringing traffic to the advertiser. This property is exploited in the click-fraud strategy studied in this paper. Figure 4.1 provides a pictorial description of the advertisement ecosystem.

To maximize revenue, ad networks often resort to *arbitrage*. At first, ad networks bought traffic only for their own advertiser customers. However, many of them now also buy traffic in order to resell it at a profit to other ad networks, either in bulk or through ad markets. This practice and the establishment of ad markets have effectively transformed Internet publicity traffic into a tradeable *commodity*. Even though publicity is an intrinsically perishable commodity, the advent of very low latency ad markets allow the same traffic (ad display or click) to be sold and re-sold many times before it lands on the advertiser's web site. As a result, the value chain between the publisher and the advertiser for the same piece of traffic can become quite long and complex.

For search engines (SE), another monetization avenue is open through *syndication*[2]. In general, syndication is the process through which a publisher integrates external content from a *syndicator* onto its web page, e.g. through an API. In the world of Internet publicity, ad networks often act as syndicators providing ads to publishers. A particular example is that of a publisher operating a search engine, where the ads are sponsored links integrated in the search results through syndication, a process called *search-engine syndication*. Thus, when the user clicks on the sponsored link on the syndicated SE, he will be redirected to the syndicator's web site before reaching the advertiser's Web page; this is necessary so that the SE can be credited accordingly.

2. This terminology is inherited from the world of electronic broadcast media.

Table 4.1 Example of a redirection chain. The first domain is the publisher and the last domain is the advertiser. The domains in between are those of the intermediary ad networks. The referer field is not changed by HTTP 300's redirections.

Position	Request	Redirection type	Referer field
1	**web-find.org/clk2?d=w4NK8...**	**HTTP 200**	/
2	web-find.org/r?q=kungfu4less&subid=...	HTTP 302	**web-find.org/clk2?d=w4NK8...**
3	**web-find.org/search?q=kungfu4less&subid=...**	**HTTP 200**	*- (unchanged)*
4	web-find.org/click?q=kungfu4less&subid=...	HTTP 302	**web-find.org/search?q=kungfu4less&subid=...**
5	88.214.241.236/click?sid=eef15...	HTTP 301	-
6	**207.244.71.165/redirect_js.php?ht_domain=web-find.org...**	**HTTP 200**	-
7	207.244.71.165/onclick.php?ht_domain=web-find.org...	HTTP 302	**207.244.71.165/redirect_js.php?ht_domain=web-find.org...**
8	207.244.71.165/local_bidding/onclick.php?affid=...	HTTP 302	-
9	adupmediaxml.com/bid_redirect.php?id_camp=...	HTTP 302	-
10	adupmediaxml.com/header_redirect.php?id_camp=...	HTTP 302	-
11	www.entrepreneur.com/topic/youve-arrived	-	-

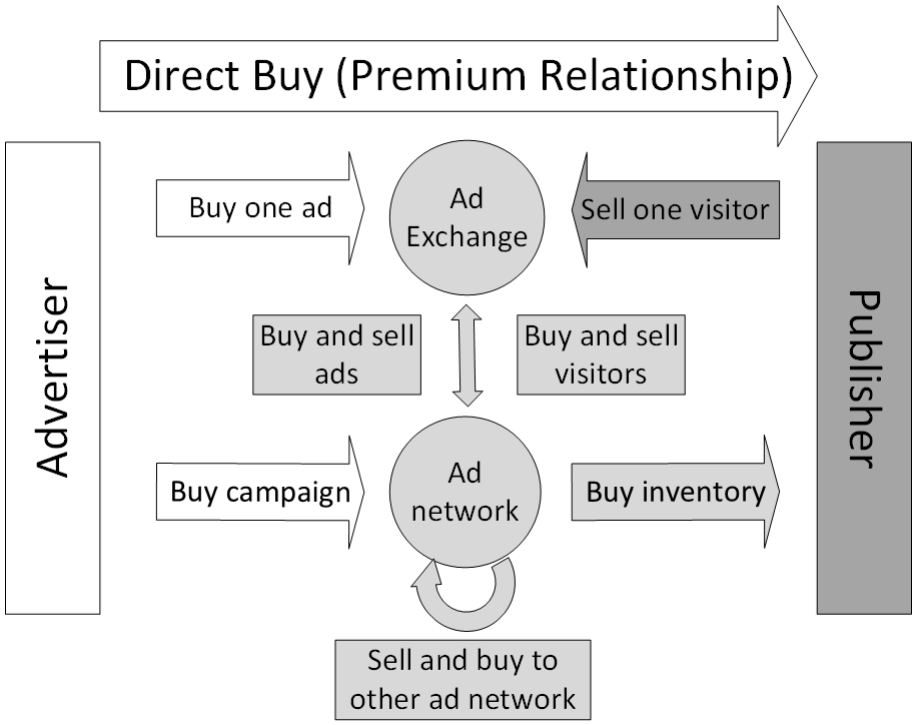


Figure 4.1 Advertisement ecosystem

This is not only true for syndicated SE, but also for all publicity traffic. Every time an ad network acquires traffic, the visitor is redirected to that ad network's Web site so that: 1) a decision can be made as to whom to sell the traffic to (an ad network or a publisher), and 2) an HTTP request that adequately identifies the ad network's account is constructed and sent to the buying party's Web site. If the traffic is subsequently resold by the purchasing ad network, a similar process of redirection will be repeated, until the traffic reaches the advertiser. We call the corresponding sequence of HTTP requests and redirections the *redirection chain*. These redirections can take multiple forms, including through ordinary HTTP 300's redirect codes, but also through JavaScript or the HTML meta tag. To illustrate this process, an example of a redirection chain is presented in Table 4.1. Some of these redirections are within an ad network's own site, but those with changing domain name or IP address typically correspond to a purchase and sale of traffic. In principle, these redirection chains can be reconstructed from the network traffic traces captured on the user's computer.

Note that there is no guarantee that all intermediaries will be visible in the redirection chain. It would be theoretically possible for ad networks to have previous agreements that do not appear in the redirection chain. However, if there is no blind trust between the actors, an ad network that is not present in the redirection chain will not be able to measure and gather

various tracking data on the traffic caused by the ad. Thus, we believe that the redirection chains contain all relevant intermediaries in the value chain between the publisher and the advertiser.

4.2.2 Ad fraud

Knowing the volume of the online ad market, it is not surprising that it has become a prime target of fraudsters. Several techniques exist to defraud both advertisers and publishers. In this paper we limit ourselves to click fraud, i.e. the automated generation of fake clicks on ads to generate fraudulent revenue.

In this kind of fraud, the primary victims are the advertisers. They are buying clicks to increase audience and brand recognition. In turn, they expect this increased visibility to translate into increased revenue, through an increase in sales for example. However, that is only true if the traffic is from *bona fide* interested human visitors. If the visitors are scripts running on infected bots, little profit will be obtained from the clicks the advertiser paid for. Nonetheless, depending on the business model of the advertiser, it may be possible for the advertiser to shift the cost of the fraud elsewhere. If the advertiser that bought fraudulent traffic is *also* a publisher, it can still display its ads to the fraudulent visitor, pushing the costs to the advertisers paying for that ad space. In that sense, if the advertiser/publisher is able to perform arbitrage between the value of its ad space and the cost of buying traffic, it may even profit from click fraud. As an example of this, Bloomberg reported in an article published in September 2015 (Elgin et al., 2015) that the Bonnier group, a bicentenary Swedish media company that recently launched several web sites, was buying botnet traffic to increase its audience and its own advertising revenue. In this paper, we will consider this phenomenon to be out of scope.

As for the ad networks, it is interesting to note that they are not always victims either. In fact, they can earn money for each click sold, as long as they receive more revenue for the click sold than for its purchase, and this holds true even if the click is fake. The exception is when the fraudulent click is detected by the downstream ad network (or by the publisher); in that case, the ad network may not be compensated for a click it actually paid for. Thus, unscrupulous ad networks can be motivated to accept as much fraudulent traffic as possible without triggering fraud detection algorithms, as was shown in the case of Yahoo in 2009 (Krazit, 2009) who was forced to settle in a lawsuit involving click fraud transiting through their ad network services. Nonetheless, Mungamuru *et al.* (Mungamuru et al., 2008) proposed that ad networks could actually benefit from aggressively fighting fraud. They argue that ad networks filtering fraudulent clicks most aggressively will have a competitive advantage, which could result

into an increased market share. The rationale behind this conclusion is that the short term incentive of immediate profits is offset by the long term loss of viability coming from displeased customers.

If the advertisers are the primary victims, and ad networks can sometimes be defrauded too, how do the fraudsters turn a profit? One obvious option for them would be to become publishers and run their own web sites to attract real users and generate traffic. But this would be tantamount to running a legitimate web site, including all the investment of effort required. On the other hand, generating fake clicks toward publisher web sites that do not belong to them would only generate revenue for those publishers and not the fraudsters. In order to capture revenues while minimizing web content creation, fraudsters capitalize on SE syndication. Since SE do not have their own content, it is easy for fraudsters to create a web site resembling an SE with minimal web content creation efforts. Using SE syndication from ad networks provides a mechanism for fraudsters operating these SE to sell traffic through them. The bots then generate "searches" on these SE, who incorporate sponsored links that can be sold as CPM or CPC. Because these SE are the entry point for fraudulent traffic, we call them *doorway search engines*. Typically, these SE only exist to lend an air of legitimacy to click fraud and have no real users. We provide a pictorial depiction of this kind of fraud scheme in Figure 4.2.

4.3 Collection methodology

In order to study the click-fraud ecosystem, we propose to observe on the activities of botnets by collecting network traces of infected machines over time. These network traces can then be analyzed to reconstruct the redirection chains traversed by these bots in their click-fraud activity. Aggregation of these redirection chains by regrouping sites belonging to the same actors allows us to reconstruct a graph of actors involved, willingly or not, in the click-fraud activities of these botnets.

4.3.1 Boaxxe

Boaxxe is the detection name for a well-known and documented click-fraud botnet (Calvet, 2014); it is also known as *Miuref*. It was first found in the wild in 2012. Boaxxe is a single-purpose botnet doing only click fraud. Unlike other botmasters who recruit machines with a *pay-per-install* model, Boaxxe's botmaster employs a network of *affiliates* who install the Boaxxe bot code on compromised machines they have infected or bought. These affiliates install the malware with a hard coded affiliate ID, which is then used by the botmaster to

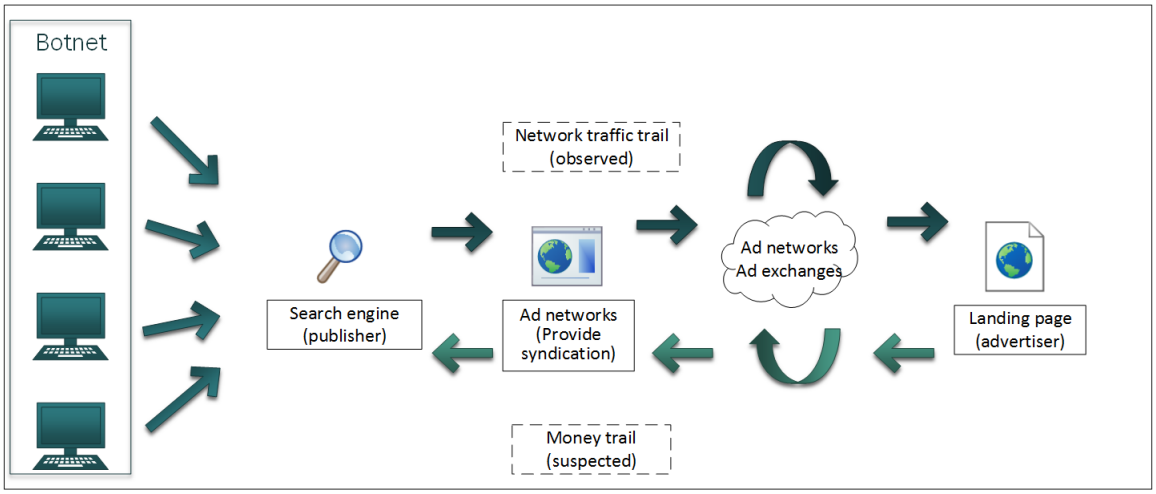


Figure 4.2 Traffic and money flows for a click-fraud scheme using a doorway search engine.

track and remunerate its affiliates for fake click traffic generated from their machines.

To perform click fraud, Boaxxe uses two modes: clickjacking and automated click fraud. In clickjacking (a.k.a. click hijacking), the malware intercepts search requests and clicks made by the real user in order to replace the target of these clicks and requests by advertisement provided by Boaxxe. In automated click-fraud, the malware simply generates traffic in the background, to make it appear as if the user of the infected machine is clicking on ads. For our study, we decided to focus solely on automated click-fraud.

In this mode, Boaxxe launches multiple click-fraud threads. Each of these threads starts by contacting a doorway SE, presumably controlled by the Boaxxe botmaster, such as `asearchit.com`, `tersearch.com` or `fesearch.com`. A screen capture of one of these SE is presented in Figure 4.3. The reply from the doorway SE is a redirection URL that contains the affiliate ID (in the `subid` variable) and a search keyword (the `q` variable). This keyword is provided by Boaxxe and is passed on to the syndicator ad network, in order to make the redirect look like a legitimate search result. Indeed, it is important to note that the choice keyword can influence the sale price of a click. When the infected computer browses this URL, it enters the advertisement ecosystem and a long chain of redirection through various actors begins. The redirection chain ends on the advertiser's web site, the *landing page*.

We believe that Boaxxe is a good representative of the automated click-fraud monetization scheme. Some investigations have indicated that other click-fraud malware, such as Pigeon, Alureon and Wowlik, also rely on doorway SE (Kalnay and Horejsi, 2014). However, the full investigation of the differences in the market surrounding the Boaxxe malware and other

Figure 4.3 `tersearch.com`

automated click-fraud modules is left for future investigations.

Unfortunately, due to the limited availability of Boaxxe samples, we had to rely on a single malware affiliate ID during the course of our study. While we have no reason to believe this introduces a significant bias in our results, this represents a limitation of our study.

4.3.2 Ramdo

Ramdo is the detection name for another well-known click-fraud malware family (Counter Threat Unit (CTU) Research Team, 2016), also called *Redyms*. It was first found in the wild in 2013. It specializes in click fraud, but can also be used to drop other malware on the infected machine.

To perform click fraud, Ramdo uses the Chrome Embedded Framework. It can be used to embed a browser in an application. Similarly to Boaxxe, it first contacts a doorway search engine, such as `search-visit.com`, and clicks on the link provided on the homepage. Then, it will be redirected, trough several ad networks, to the landing page. A screen capture of the search engine is presented Figure 4.4.

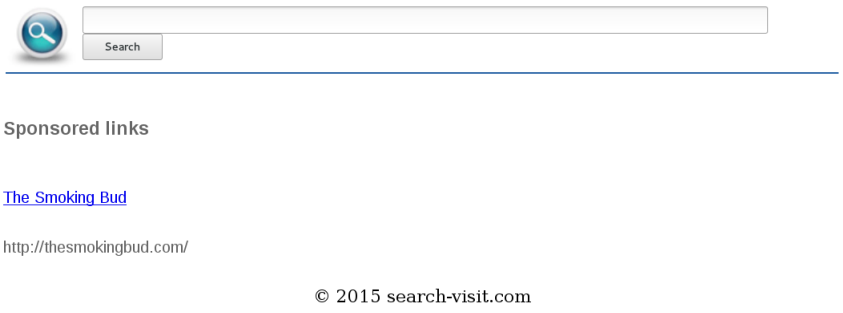


Figure 4.4 `search-visit.com`

In order to collect Ramdo click-fraud chains, we used an instrumented and scriptable browser, *PhantomJS* (Hidayat, 2016), to crawl the doorway search engine. As the link used by Ramdo to perform click fraud is provided on the homepage, we scripted the browser to 1) contact the SE, 2) click on the link, 3) intercept and follow the redirections, and 4) when no further redirections appear within fifteen seconds, we save the redirection chain in a text file. As a result, the redirection chains collected are directly usable for analysis.

4.3.3 Longitudinal study

Knowing the advertising market is composed of a series of campaigns, each lasting a limited period of time, we chose to make a *longitudinal* study. This means, that data is collected frequently and over a long period of time. For this study, two different datasets were collected.

The first dataset was collected at ESET using their internally developed Windows sandbox infrastructure. The data collection spanned seven months, from April 2015 to the end of October 2015. During that period, every day a virtual machine was infected with Boaxxe and let to run for a period of ten minutes for the first two months, and for thirty minutes subsequently. For each of these runs, a complete network trace was collected.

The second data set was generated at École Polytechique de Montréal for a period of one month in January 2016. Each infection was run for thirty minutes on a virtual machine image containing only a freshly installed Windows 7. Unlike in the ESET dataset, a new virtual machine was launched immediately after the previous one was terminated, with data collection happening 24/7 during that month. The network traffic to and from these virtual machines was saved using the *nictrace* VirtualBox option. An anonymizing network was employed during this experiment, with the geographical location of the exit points being regularly changed, in order to measure the potential influence of location on our results. Since no additional software was installed, no additional HTTP traffic was present in the network traces, thus avoiding the need for a filtering step in the processing of traces.

Table 4.2 Data summary of the Boaxxe longitudinal study. The external redirection count includes only those transiting from one domain to another.

	Boaxxe	**Ramdo**
Total size (PCAP)	3.8 GB	NA
Duration (days)	207	54
Number of chains	1380	9596
Number of external redirections	3218	32369

The control experiment based on this second dataset appears to confirm that the results collected in the longitudinal study of the first dataset are valid. The longer daily collection greatly increased the quantity of data collected, but the large majority of that data was redundant in terms of actors identified and their relationships. However, geographical location did seem to have an impact on results. Notably, in some non-English speaking countries, no automated click-fraud activity could be observed. Conversely, other activity not related to click-fraud was observed for United States IP addresses. We did not determine the nature of this traffic, but we suspect it might have been related to search-engine optimization (SEO). Nonetheless, the automated click-fraud activity, when present, was consistent. This would suggest that, while some differences based on geographical location were observed, they do not detract from the generality of our observations related to automated click-fraud for Boaxxe.

To validate our methodology, we used a third data set as a control experiment. It consists of click-fraud chains collected from a doorway search engine of the click-fraud malware Ramdo. These chains will be used to validate the chain reconstruction methodology as they were collected using an instrumented browser. Moreover, this dataset will be used to validate the reliability of the disruption methods comparison, detailed in section 4.5. As the Ramdo study is not a full longitudinal study, we will use these data only for validation and not to compare the ecosystems. A summary of the first and the third dataset is given in Table 4.2.

4.3.4 Chain reconstruction

As seen in Section 4.2.1, we can use the redirection chain to act as a proxy for the value chain of the advertisement ecosystem. Unfortunately, the data collected from Boaxxe is raw packet capture files (pcap). It is necessary to extract the HTTP redirection chains to study click fraud. While it is easy to extract individual HTTP requests from a pcap file, it is difficult to link the individual HTTP requests to specific advertisement redirection chains. This is mainly due to the number of redirection chains that occur at the same time in simultaneous threads, and to the variety of redirection types observed in advertisement chains. In particular, a number of redirections are dynamically generated by JavaScript. Figure 4.5 is an example of a redirection URL that depends on the timestamp. Moreover, because of the presence of HTTP 300's redirections, it is not possible to blindly trust referer information. For example, the third column of Table 4.1 shows that HTTP 300's redirections did not modify the Referer field.

In order to solve these problems, we developed an algorithm to reconstruct the redirection chains. Basically, the algorithm parses the content of each HTTP packet to retrieve URL.

```
var t = Date.now();
var url = "http://www.website.com"
+"/p.php?t="+t;
//http://www.website.com/p.php?t=timestamp
window.location.href = url;
```

Figure 4.5 Example of a dynamic JavaScript redirection

Then, using information gathered from the URL, each HTTP request is linked to the HTTP response which triggered the redirection.

As we statically retrieve the URL, it is not possible to match the request with the previous response if the redirection URL was dynamically generated in JavaScript. In most of the scripts manually studied, the script only feeds parameters in the URL and rarely adds new parameters. Thus, we have chosen to use the following distance metric:

$$d(u_1, u_2) = \frac{\#\text{different parameters}}{\#\text{parameters}}$$

If the parameters and the host are similar, the distance is low and the two URL are likely to be the same. As a last resort, we rely on the HTTP referer field to find the previous HTTP response.

On the other hand, multiple click-fraud threads can run at the same time and are possibly browsing exactly the same URL. To address this problem, we chose the most recent request if the time difference between the two most recent ones is above one second. Indeed, Crovella *et al.* showed that the OFF times, the time difference between a response and a request, has a probability of 85% to be under one second (Crovella and Bestavros, 1997). Based on these heuristics, we built the algorithm presented in Algorithm 1.

This generates a tree for each thread, where the root is the initial request to the doorway SE. The nodes of this tree are the URL that were subsequently requested, with each edge representing such a request from the parent URL to the child URL. These requests include the advertisement redirection chain, but also requests to auxiliary resources, e.g. images, CSS, page counters, etc. Figure 4.6 is an example of such a tree.

Once the tree is built, we then extract the advertisement redirection chain. It is simply the path between the root of the tree and the landing page. To find this path, it is necessary to identify which node corresponds to the landing page.

Since advertiser landing pages typically provide rich content, they contain many requests to

```
Data: NT:NetworkTraces
Result: G:Graph
Initialize url_list;
for each reply A in NT.A do
    Search URL in A's data;                                                    //
                                  // All redirection types are taken into account
    Stores URL in url_list;
end
for each Request R in NT.R do
    continue ← True;
    if R.url is in url_list then
        Replies ← getRepliesFromUrl(R.url);
        if Replies.length > 1 and Replies[1].date - Replies[0].date > 1 then
            Replies[0].next ← (R,0);
            continue ← False;
        end
    end
    if continue is True then
        dist ← +∞;
        umem ← Null;
        for each u in url_list do
            computeD ← d(u, R.url);
            if computeD < dist && computeD < 0.6 then
                dist = computeD;
                umem = u;
            end
        end
        if umem is not Null then
            if Replies.length > 1 and Replies[1].date - Replies[0].date > 1 then
                Replies[0].next ← (R,dist);
                continue ← False;
            end
        end
        if continue is True then
            if R.referer is in url_list then
                Replies ← getRepliesFromUrl(R.url);
                if Replies.length > 1 and Replies[1].date - Replies[0].date > 1 then
                    Replies[0].next ← (R,Referer);
                end
            end
        end
    end
end
```

Algorithm 1: Reconstruction of redirection chains

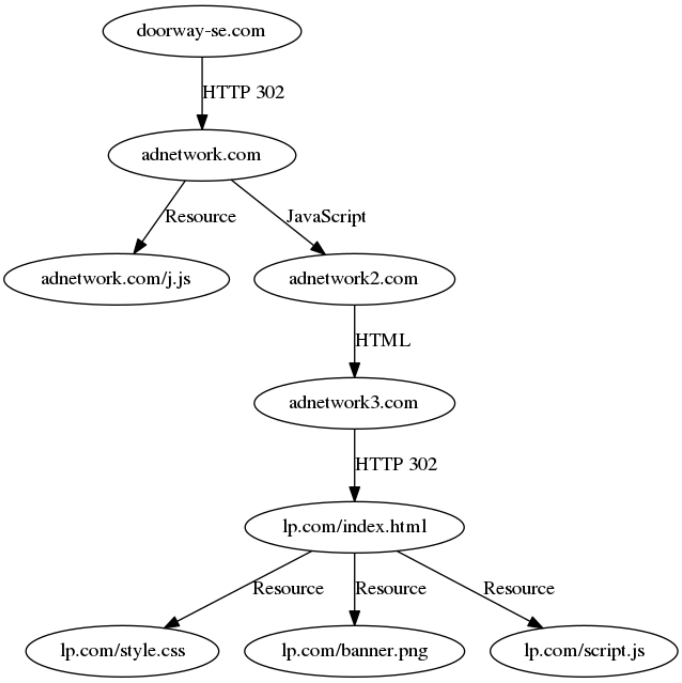


Figure 4.6 Example of a HTTP browsing tree

internal and external resources. In contrast, we observed that intermediary nodes (corresponding to ad networks) generally only contain redirection scripts and few or no resources. This allows us to identify landing pages relatively easily by counting the number of children.

Using several chains from the Ramdo dataset, we were able to validate our reconstruction method. Thus, we compared 150 Ramdo chains from the instrumented browser to the same chains reconstructed from the network file (PCAP). It turns out that all the chains were properly reconstructed. However, for 25 chains (17%), the landing page was different; the chains from the instrumented browser were longer, even if the beginning of the chain was strictly equal. This may be explained by the difference in the definition of a landing page. For the reconstruction method, the landing page is the node with the most children while for the instrumented browser, the landing page is the last page if no further redirections appear within fifteen seconds. Thus, it is possible that a traditional web site, identified as a landing page by the reconstruction method, performs a redirection that is caught by the instrumented browser. This redirection can be caused by a malicious ad or by the web site owner. In this case, it means he is able to perform arbitrage between the price he bought the traffic at and the price other advertisers or ad networks are willing to pay. However, this distinction in the definition of the landing page has little bearing on our results because the main focus of our study is in the ad networks that are acting as intermediaries in the

click-fraud value chain.

Once all the redirection chains are reconstructed from the various trees, we merge the chains into a single graph by regrouping all nodes that share the same domain name or IP address. In that graph, the nodes are the domain names or the raw IP addresses extracted from the redirection chain nodes. The presence of an edge between two nodes means that a redirection between these nodes was found in at least one chain.

4.3.5 Node aggregation

This graph cannot be considered an accurate depiction of the business relationships between actors involved in Boaxxe click fraud, because actors typically operate several IP addresses and domain names. In order to make it more useful, we would prefer if the nodes represented actual actors. Thus, we should merge all the nodes belonging to the same organization into a single node. To do so, we developed a methodology based on Whois data, passive DNS data, tracking codes and page similarity.

First, we collected the data for each web site from the Whois database to gather the registrant's name, address, email address and phone number. We also gathered the authoritative name server when it was not a registrar or hosting service. Two web sites registered to the same company, at the same address and using the same email are likely to belong to the same actor. However, special care must be taken when automating this process because many web sites, including legitimate web sites, use Whois anonymizer services, especially to protect their email address, which can lead to inappropriate merging of nodes.

Second, we used the Virus Total passive DNS service (Virustotal, 2015) to retrieve the IP addresses resolved by each domain name. Two domain names resolving to the same IP may be an indication that they belong to the same actor. Again, care must be taken because of shared hosting or denial-of-service protection services. Thus, we also verified that the IP address did not belong to a known cloud provider like Amazon EC2 or a DoS protection service such as CloudFlare. In some cases, we also considered the DNS Start of Authority (SOA) record because it contains not only name servers, but also an email address.

Third, we parsed the index file of each web site to retrieve tracking codes, which are account numbers for affiliate programs, a technique developed by Seitz (Seitz, 2015). Because these codes are used to produce sensitive information on page counts (e.g. Google Analytics) or to attribute revenue (e.g. Google Adsense), they are unique and are not normally shared across organizations. As such, they are a good indicator that a node belongs to a particular actor. We chose to use the five following affiliate programs: Google Analytics, Google Adsense,

Amazon, ClickBank and AddThis.

Finally, in some cases, we noted that several web sites shared the same web page in which only the domain name was changed. By looking at the source code of these web pages, we could confirm that they were indeed identical. We considered only original web pages to avoid default configuration pages. Similarly, we found that several web sites shared the same SSL certificate. While normally web sites with different domain names should not share a certificate, we found that some sites use the same SSL certificate when we crawled them with HTTPS. This is a good indicator that the same default template, including the SSL certificate, was probably used in constructing these sites. Thus, their reuse suggests that they belong to a single actor.

To ensure the quality of this process, we performed a manual check of each merge. In other words, we collected data automatically to support node merging, but we manually confirmed each merge. Overall, this aggregation process allowed us to reduce the number of nodes in the initial graph from 523 to 225 potential actors.

The resulting *actor graph* is the graph where all redirection chain are aggregated and all nodes are merged as described above. We believe that this graph represents an adequate sampling of the overall business relationships between actors in the Boaxxe ecosystem, i.e. *who* is doing business with whom. In addition, we can calculate the weights for each edge based on the number of redirection chains that transited between the corresponding merged nodes. However, due the limitation of our sampling collection methods, we cannot claim that this weighted actor graph gives an accurate description of volume of Boaxxe click fraud, nor on the relative importance of these relationships in terms of fraud revenue and cost. In other words, we cannot quantify *how much* business is being done between the actors.

4.4 Actor graph analysis

In this section, we present the results of analysis of the actor graph. We will first give an overview of the graph. Then, we will present some actors of the graph. Finally, we will give insights on the evolution though time of this ecosystem.

4.4.1 Actor graph

Using the chain reconstruction process presented in Section 4.3.4 and the node merging procedure described in Section 4.3.5, we reconstructed the actor graph of Boaxxe's automated click-fraud ecosystem. The resulting actor graph is depicted in Figure 4.7. The visual representation of this graph in Figure 4.7 is generated by the *twopi* software, which uses a radial

layout proposed by Wills (Wills, 1997). It is a tree-like layout in polar coordinate. This means that the algorithm places nodes on a planar surface, with the doorway search engine as the root of the tree. The nodes at the same distance to the root are on the same radius. Thus, this will place most of the ad networks near the center of the representation while the landing pages will be pushed towards the periphery. We chose this representation because it allows to visually identify the role of each node and its importance in the ecosystem. As this is a tree-like layout, it is possible to identify which portion of the graph is controlled by a single actor or a group of actors.

As a result of the node aggregation procedure, all domain names for the doorway SE were merged into a single node, which we call the *Boaxxe root* or simply root as it has no incoming edges. By analyzing the graph data, we observe that immediate neighborhood of the root is relatively small. The components directly linked to the Boaxxe search engines (neighborhood of radius 1) represent only 5.36% of the nodes in the actor graph. The small number suggests that they may become a potential choice for disruption. Graph density is the proportion of edges present in a graph in comparison with a fully connected graph of the same size. In our case, the relatively low graph density of the radius-1 neighborhood (0.348) suggests that disruption operations are tractable by targeting a limited number of these nodes.

Another observation from the data is that the nodes of the network are regularly reused. Specifically, 58% of the nodes were visited more than once. The average number of times a node was visited is 13 times, with a standard deviation of 40. The fact that these nodes are visited multiple times even with our limited sampling suggests that these relationships are long-term commitments and not casual relationships.

Of the 225 nodes of the graph, 113 were landing page actors and 11 were in the immediate neighborhood of the Boaxxe root. It is important to note that all actors in the immediate neighborhood were identified after only 73 days of data collection, suggesting than our sampling probably provides sufficient coverage of the key intermediate actors in the Boaxxe click-fraud ecosystem. On the other hand, new landing pages were discovered regularly for the duration of the study. However, this is the expected behavior because of the volatile nature of their advertising demand (i.e. campaign-driven advertisement). This has no bearing on our results because the landing pages are not intrinsically part of the criminal ecosystem.

4.4.2 Actors

Because Boaxxe's click-fraud scheme is organized around a syndicated doorway SE, it is useful to delve more deeply in the ad networks that are offering it syndication. After all, it seems fairly evident that Boaxxe is involved in malicious activity. Even if the search

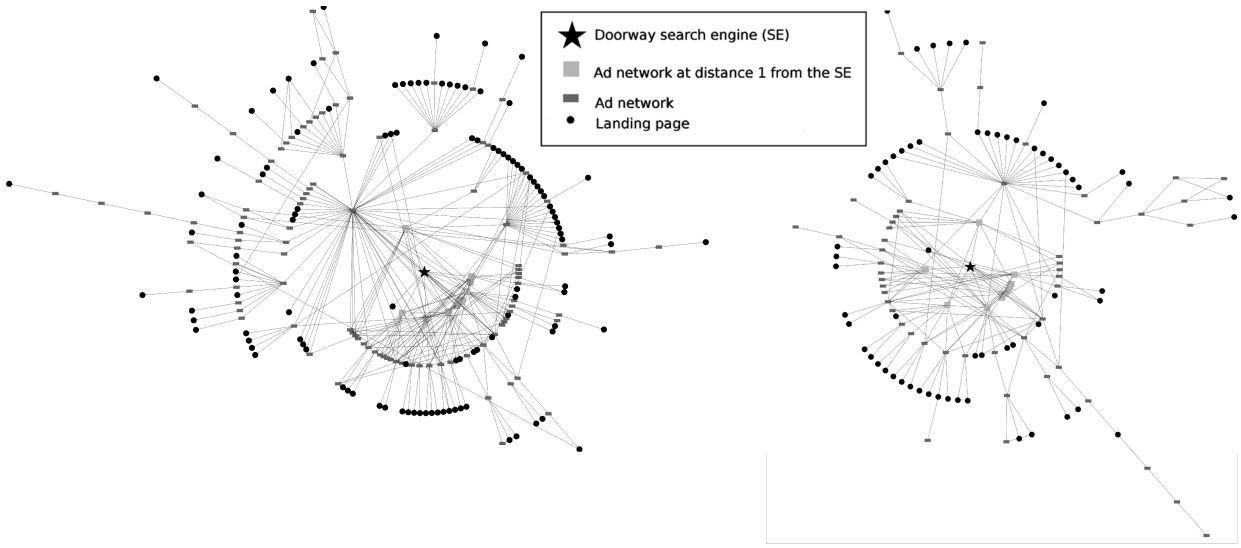


Figure 4.7 Comparison of the actor graphs (in Wills radial representation) of the non-disrupted (left) and disrupted (right) click fraud ecosystems. The most central node is the doorway search engine and the other nodes are on the circle of radius corresponding to their distance to the search engine. The disrupted graph is the result of removing three key players. Note that the connected components of the disrupted graph that are disconnected from the doorway search engine have been removed from this depiction.

engines used by Boaxxe regularly change domains, they always use the same HTML page. Furthermore, any deep inspection of the traffic would reveal that the traffic is generated by bots. For instance, the search results obtained from the doorway SE have nothing to do with the search requests. As such, it is reasonable to suspect that the ad networks directly linked to the doorway search engines are buying botnet traffic knowingly.

We can look at some examples of ad networks directly connected to the Boaxxe SE, i.e. at a distance of 1, to understand this category of actors. One of these ad networks is called *Nextadnet*. Based on the Whois information of web sites owned by Nextadnet, we were able to determine that this company is based in Cyprus. However, when browsing their web site, the only methods offered to contact them are via Skype, e-mail, ICQ and Jabber, which is unusual even for an Internet-based company. Moreover, someone claiming to be a representative of that company posted an offer to buy traffic on *blackhatworld* (Blackhatworld.com, 2015), a forum of questionable reputation, known for providing information and tools for Black Hat SEO techniques. It would be surprising if that was done with the intent of acquiring good-quality traffic for their advertisers.

Another example is the *superior-movies.com* web site, which was regularly used as the first redirection after the doorway SE during the first four months of our study. Its homepage

consists of several movie trailers, none of which refer to recent movies. However, if the web site is browsed with a particular set of URL parameters, it will redirect the user to an IP address of the media company *Daoclick* instead of landing on the homepage of `superior-movies.com`.

These two examples paint the portrait of typical fly-by-night advertisement companies that knowingly deal with illicit or unethical actors.

In comparison, the ad networks with a distance of two from Boaxxe are a mix of well-known media companies, like `advertise.com` or `ad.com`, domain parking services, such as *Parking Crew* or *Go Daddy parking*, and ad networks deliberately and publicly offering low quality traffic, like `popcash.net`. For the most part, these are well known companies with a presence in the legitimate market.

Looking at the landing pages, we observe well-known web sites, like Amazon, Bing or the Huffington Post, but also suspicious web sites like `fasttcash.biz` that proposes get rich quick schemes, or `valortechhelp.com`, a web page containing nothing but ads. Interestingly, we also found, within the landing pages in the first months of our study, the same Bonnier group mentioned in the Bloomberg article cited in Section 4.2. We can also confirm from our dataset that the traffic received by Bonnier during that period came from `advertise.com`, as was also described in the Bloomberg article. Overall, we found that 12% of the landing pages in our dataset were part of the Alexa top 10,000. Another 29% are out of the top 10,000, but are still within the Alexa top 1,000,000. The fact that 41% of all observed Boaxxe traffic ends up in Alexa top 1,000,000 web sites, most of which are presumably legitimate web sites, strongly underlines the fact that Boaxxe traffic has no difficulty entering the legitimate advertisement market.

4.4.3 Evolution through time

It is interesting to look at the evolution of the different actor groups over the duration of our study.

Figure 4.8 shows the normalized distribution of the number of days between the last appearance and the first appearance of a node in our dataset. In the figure, we compare the distribution from landing pages and from ad networks directly connected to doorway search engines. We can see that landing pages typically only appear for a short period of time, as could be expected from an advertisement campaign model. In comparison, the ad networks connected to the search engines were much more persistent, with around half of them appearing for longer than 120 days. Thus, the actors near the doorway search engine were stabler than the landing page actors.

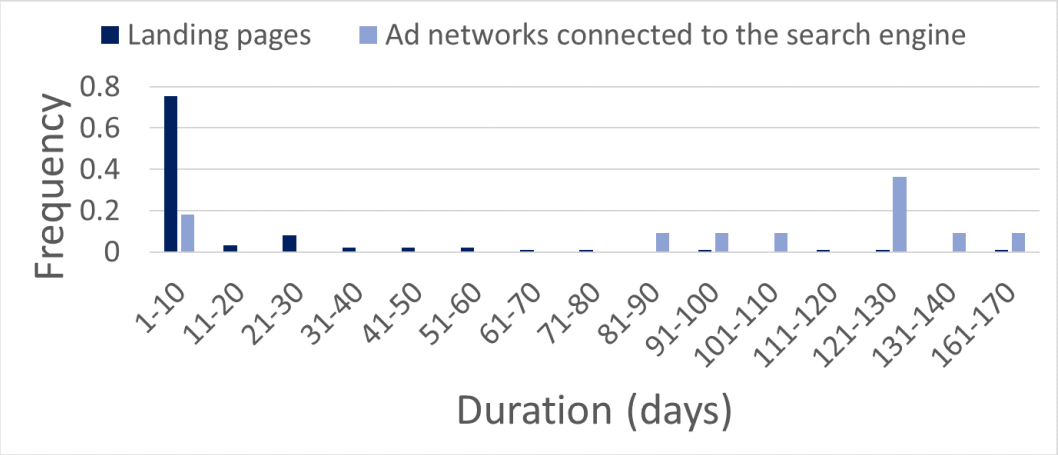


Figure 4.8 Normalized distribution of the number of days between the first and last appearance of nodes in our dataset for landing pages nodes and ad networks nodes at a distance of 1 from Boaxxe.

4.5 Disruption approaches

While analysis of the actor graph may provide insight on who might be involved in the distribution of fradulent traffic, willingly or or not, it does not provide by itself a clear solution on how to reduce the impact of click fraud on the legitimate advertising business.

In order to disrupt click-fraud operations, direct action targetting the malware operators and their affiliates directly is not really practical. First of all, it is hard to determine who they are exactly in order to initiate criminal prosecution or other kinds of legal action. Second, there is no simple and straightforward way for an intermediate actor to immediately identify *a posteriori* that a particular piece of traffic is fraudulent once it has gone through a couple of redirections. Thus, breaking the value-chain of click fraud by simply refusing fraudulent traffic is not an option.

As we discussed in Section 4.2.2, detecting fraudulent traffic *after the fact* is not necessarily that hard if *a posteriori* analysis can be performed on publicity traffic patterns and trends. Theoretically, if all actors withheld payment of accepted traffic long enough until such analysis could be performed, then fraudulent traffic could be picked out and not paid for. Indeed, this is one of many ways in which the banking industry controls credit card fraud: charges to credit card are paid immediately but can be "charged back" for a period of 30 days. In the current world of Internet publicity, however, this is not standard practice. For many intermediary actors this is not attractive. First, because they would have to bear the cost of deploying and operating such detection mechanisms. Second, there would be the cost and business impact of having to deal with complaints from unpaid providers, including loss of

and higher prices of traffic, damage to reputation and legal costs. Lastly, there might not a business incentive to do so if the traffic (even if fraudulent) can be readily sold to unaware customers downstream.

This constitutes another example of the typical impasse in Computer Security where those that could make a difference by changing behaviour are not immediately incentivized to do so. In fact, individual change by a good willed actor, e.g. by starting to filter or defer payment for suspected fraudulent traffic, might actually result in an immediate loss for that actor, unless the other actors change their behaviour as well, since they could pick up that same fraudulent traffic and make a profit from it. Simultaneous change by all actors in the ecosystem would very hard to coordinate and is unlikely to be unachievable. Therefore, we must consider disruption strategies that are effective if only a limited portion of the actors can be swaid to change behaviour. Accordingly, it becomes important to identify which actors play a more central role and are therefore the most important targets in the disruption effort.

In the rest of this section, we describe and discuss other disruption scenarios with similar objectives of identification of critical actors: takedowns in peer-to-peer botnets and disruption of criminal networks.

4.5.1 Disprution of peer-to-peer botnets

The rise of botnets employing peer-to-peer (P2P) communications as a command and control mechanism in the mid 2000's led to research efforts on how to disrupt their operations. In P2P botnets, each infected machine knows the addresses of a number of other infected machines, information that is contained in its *peer list*. This peer list is then employed by the P2P communication algorithms to route command and control messages to and from the botmaster and the infected machines. The graph constructed by considering the infected machines as nodes and relationships in the combined peer lists as edges is indeed akin to the actor graph of click fraud.

A first approach to disrupting botnet operations consists in by altering the behaviour of an infected node, e.g. by disinfecting in. This can be viewed as a *node removal* strategy, because even though a disinfected node could still remain in another machine's peer list, it will no longer participate in communications. Initial research by Davis *et al.* (Davis et al., 2008a) evaluated the efficacy of different node removal strategies in terms of various criteria, such as the number of disconnected nodes or the proportion of nodes that could be reached within a fixed number of hops within the network, all the while constraining the disruption efforts to a maximum number of removed nodes. Various random graph models, such as the Barabási-Albert and Erdös-Rényi graphs, were used to emulate the peer-to-peer graphs in

botnets. The strategies proposed and evaluated were meant to simulate different scenarios in which the disrupting actor would have access to different resources and tools. For example, in the *random strategy* nodes to be removed are chosen randomly, representing a scenario in which the disruptor has minimal information on the actor graph. In the *tree-like strategy*, the disruptor has local information on the peers of a particular node, i.e. the peer list, and can then remove this node and subsequently its peers. This is is done recursively until a maximum number of nodes has been removed. In contrast, the *global strategy* employs full knowledge of the actor graph to target those nodes that have a larger degree (i.e. importance) in priority, in order to achieve maximum disruption. In the case of botnets and the random graphs generated by typical P2P protocols, their work showed that not much advantage could be gained from having access to global knowledge on the most "popular" (most connected) nodes in the graph.

An alternate approach to node removal is that of *Sybil attacks* or *poisoning attacks*, in which rogue nodes are introduced into the network to disrupt its activities. The efficacy of such techniques in the context of P2P-botnets has also been studied (Davis et al., 2008b) and implemented experimentally (Calvet et al., 2010). This has been the strategy of choice employed in real-life takedowns of P2P botnets, since it is technically simpler and less legally and morally questionable to tamper with machine belonging to unknown owners, even if it is to disinfect them of malware. Unfortunately, this approach is not viable in the context of click fraud disruption. Insertion into the actor graph and poisoning operations would require both 1) knowingly buying fraudulent traffic, which would be expensive and 2) knowingly reselling some of that traffic, a would be morally questionnable proposition.

While previous work on P2P botnet disruption suggests and compares various approaches to disrupt graphs with limited resources, e.g. when total simultaneous disruption is not possible, the results obtained are not applicable in our cases. First of all, the underlying graphs are large and different in structure to actor graphs in click fraud. In particular, they are unstructured and no particular node represents the main malicious actor. In the case of a P2P botnet, any node can be used by the botmaster to seed a new command to the whole botnet; the graph is homogeneous and has no "center". In our case, however the initial search engines are unequivocally associated with the botmaster and our disruption objective is more specific: isolating it from the other nodes.

4.5.2 Disruption of criminal networks

Several criminologists have used network (graph) analysis metrics to study potential disruption strategies of criminal networks. These are essentially node removal strategies, where

removal would typically be achieved by arrest and prosecution or by otherwise convincing actors no longer to interact with the other actor in the graph.

Clayton *et al.* (Clayton et al., 2015) argue that poorly targeted disruption operations allow for quick recovery by criminals. Thus, it is critical to use the appropriated method to select the targets for disruption in order to achieve maximal effects. These disruption strategies can be divided in two groups: finding the best set of target for a given number of nodes removed and finding the best targets for a given number of advertisers disconnected from the search engine.

Fixed effort

The aim is to find, for a given number of actors to remove, the set of nodes that maximize the number of landing pages disconnected from the doorway search engine. On one hand, several metrics exist in network analysis to characterize the importance of each node in a graph (Everton, 2012); it is called *centrality*. Among them, we chose to focus on strategies that optimize with respect to the following metrics:

Degree. The degree is the number of edges of each node. In other words, it aims to calculate the importance of a node as a higher degree means it has a relation with a higher number of actors.

Betweenness. The betweenness of a node is the number of times a node is in the shortest path between any two nodes of the graph.

Closeness. The closeness of a node is the multiplicative inverse of the sum of the shortest paths to the others nodes of the graph. In other words, a node with a low closeness value can reach the others nodes of the graph through a small number of intermediaries.

Fragmentation. The fragmentation of a graph is based on the number and size of connected components of the network. A normalized fragmentation value of 1 means that all the nodes are disconnected. On the other hand, a fully-connected network would have a fragmentation value of 0.

In particular, fragmentation is used in the Keyplayer-Negative problem (KPP-Neg) (Borgatti, 2006), a technique shown to be effective when dealing with cyber criminal networks by Décary-Hétu and Dupont (Décary-Hétu and Dupont, 2012).

Fixed disruption level

In our case, and as discussed in Section 4.5.1, not all nodes are equal and we particulary interested in severing links between the clear perpetrator (doorway search engine) and the

clear victims (landing pages). The aim of this second group of strategies is thus to find the smallest set of nodes that will disconnect the landing pages from the doorway search engine. This is a specific subcase of the *min-cut problem*, where one seeks to find a minimal set of nodes to remove that will separate two given nodes, i.e. transforming the initial the graph into two disconnected components, each containing one of these nodes. A reduction technique that can be used to tranform our specific problem into a min-cut problem instance (solvable by standard graph software) is to define a super node. We merge all the nodes of the landing page set into a super node. Thus, we can then compute the min-cut method between the doorway search engine node and the super node, corresponding to the set of landing pages.

However, it might not be necessary nor efficient to attempt to sever links between the doorway search engine and *all* the landing pages. Thus, we consider a variant of the problem where we only want to isolate the doorway search engine from a given *percentage* of the landing pages.

4.6 Disruption results

In the previous section, we presented the different strategies that can be used to select the best actors for a disruption operation. The global strategy employed in P2P botnet disruption approaches is in fact equivalent to techniques proposed by criminologists that optimize for degree; they both take advantage of global knowledge of the actor graph to optimize disruption in a context of limited resources. In the rest of this article, we concentrate on the latter work as it is more comprehensive in terms of diversity of techniques and performance criteria.

As the Boaxxe ecosystem is represented by our actor graph, it is possible to evaluate the effect of eventual disruption operations on the actor graph. This can be done by 1) counting the percentage of landing pages that are disconnected from the doorway search engine, or 2) by counting the minimal number of actors that need to be removed in order to disconnect a given percentage of landing pages.

4.6.1 Fixed effort

In Section 4.5.2, we presented a group of methods that can be used to rank the nodes and select the best target for disruption. In order to compare these methods, we rely on a single metric: the percentage of landing page nodes disconnected from the search engine node. To compute the KPP-Neg problem, we used `Keyplayer2`, a program developed by Borgatti (Borgatti). It implements a KPP-Neg solver, with a choice of three heuristics. We

chose to use the Greedy heuristic so that results could be obtained in a timely fashion. We opted for 5,000 iterations to improve the results. On the other hand, we used a random strategy to benchmark the other methods. We randomly chose sets of ad network nodes of given sizes and calculated the percentage of landing page nodes disconnected from the search engine node. We repeated this process 10,000 times. The results are given in Figure 4.9a.

First, the Keyplayer method, which relies on fragmentation, performs better than the other methods. From 3% of ad networks removed, its curve is above the others and there is a difference of fifteen points at 10% of ad networks removed.

Second, the degree and the betweenness centrality methods are really close. At low proportion of removed actors, the degree method is performing better than the betweenness method but it plateaus more rapidly. Thus, there is no significant difference between these two methods. However, the random method is relatively close to these two methods. As such, we cannot retain these two strategies as they do not give better results than having a no-strategy (random method).

Third, the closeness centrality curve is much below the other curves. For instance, at 10% of ad networks removed, the difference between the closeness centrality and the Keyplayer is nearly thirty points. Moreover, as for the degree and the betweenness centrality, this strategy does not give better results than some results of the random strategy.

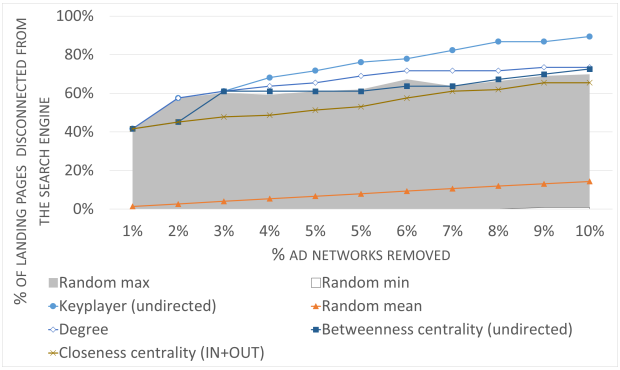
For this first group of strategies, it turns out that only the Keyplayer method may be suitable to solve our problem; it performs better than the other methods and is the only method that always outperforms the random strategy.

4.6.2 Fixed disruption level

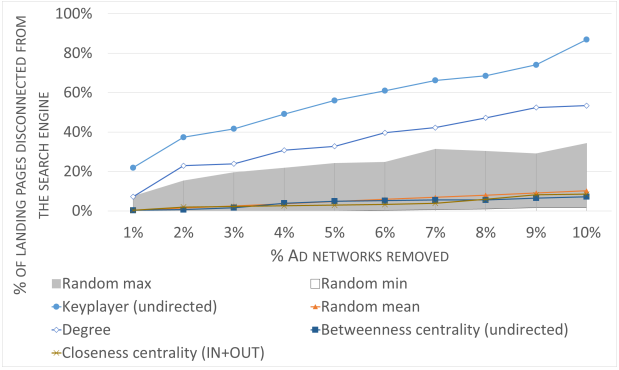
In Section 4.5.2, we proposed to use the min-cut method to find the smaller set of nodes that, when removed, disconnects a set of landing page nodes from the search engine node.

Given the relatively large number of landing pages, it is not possible to measure performance over all possible subsets of landing pages representing a given percentage of them. Indeed, the Boaxxe ecosystem contains 113 landing page actors, which represents a total of $3.7 * 10^{32}$ possible subsets of nodes of containing 50% of the landing pages. Therefore, we chose randomly 10,000 different sets of landing page nodes of the given size and compute the min-cut on them. The exception is of course for the case where we measured performance for isolation of 100% of the landing pages. The results are presented in Figure 4.10a.

By looking at the graph, we can see that the curve plateaus rapidly. For a higher than 30% percentage of landing pages disconnected from the search engine, it will necessitate to



(a) Comparison for the Boaxxe ecosystem.



(b) Comparison for the Ramdo ecosystem.

Figure 4.9 Comparison of different methods to select the best targets for disruption in the a) Boaxxe and b) Ramdo ecosystems. The x-axis is the percentage of ad networks removed. The y-axis is the percentage of landing page nodes disconnected from the search engine node. The shaded area is the area between the min and the max values of the random strategy.

remove around 10% of ad networks. This percentage increases by less than two points to totally disconnect the landing page nodes from the search engine node.

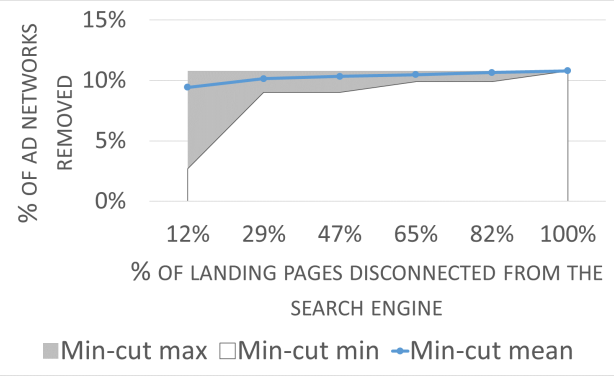
To disconnect 30% of the landing pages using the Keyplayer method, only 1% of ad networks need to be removed in comparison to the ten percent of the min-cut method. However, to fully disconnect the graph, the Keyplayer method is more costly, requiring an additional 0.2% point of actors removed. Thus, the Keyplayer technique seems more effective if we want to partially disrupt the graph and it gives similar results for the full disruption.

4.6.3 Verification

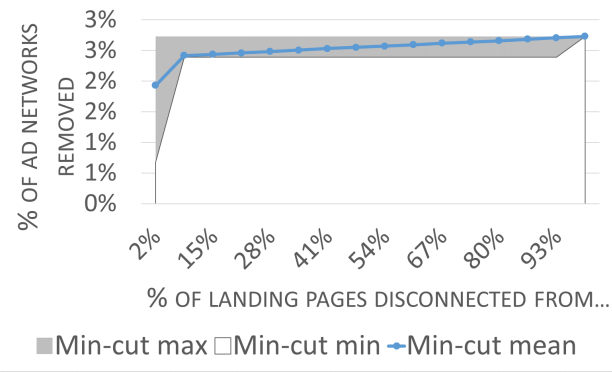
In Section 4.6, we found that the best strategy to maximize the disruption of the Boaxxe ecosystem is the Keyplayer method. However, the results of the comparison should not depend on the structure of the ecosystem. Thus, we performed a validation with data from the Ramdo malware to verify the robustness of our strategies. The results of this validation are presented in Figures 4.9b and 4.10b.

Firstly, the Keyplayer results are close to those obtained using the Boaxxe ecosystem. The curve is well above the other strategies and especially the no-strategy, with at least twenty points of difference.

Secondly, the degree method no longer tracks the closeness and betweenness centrality strategies which was the case in the Boaxxe ecosystem. This illustrates the lack of robustness of these strategies as changes in the ecosystem structures induce large variations in the perfor-



(a) Results for the Boaxxe ecosystem.



(b) Results for the Ramdo ecosystem.

Figure 4.10 Results of the min-cut method on the a) Boaxxe and b) Ramdo ecosystem. The x-asis is the percentage of landing pages to be disconnected. The y-axis is the percentage of ad networks to be removed to disconnect the given set of landing pages.

mance.

Thirdly, the betweenness and the closeness centrality strategies are totally ineffective. They are below the no-strategy and really far from the Keyplayer and the degree strategies.

Finally, the min-cut curve plateaus more rapidly, around ten percent of landing pages disconnected.

As such, this validates the results of the comparison of the different strategies using the Boaxxe ecosystem.

4.6.4 Interpretation

In the section 4.6.1, we showed that, contrary to the Keyplayer technique, most of the traditional network analysis metrics are not appropriate to our problem: maximizing the number of landing page nodes disconnected from the search engine node. Thus, fraudulent traffic could still be channeled through other actors and redirection paths even if the highest centrality ad network is "removed". As long as alternate redirection paths exist, fraudulent traffic can be sold. The ecosystem may offer less flexibility to the fraudsters, but it will remain able to accomplish its purpose: monetize fraudulent traffic. In particular, targeting ad networks with high betweenness scores may lead to inefficient removal of actor. Closeness metrics capture the criticality of edges in terms of how graph distances are affected when they are removed. In the case of click fraud, an increase of distance due to the disruption of high betweenness actors would only force the fraudulent traffic to transit through more intermediaries to reach the same landing pages. This would hardly constitute an important

disruption to monetizing operation in most cases.

On the other hand, in the section 4.6.2, we showed that the min-cut method may be efficient to select the intermediaries that, when removed, will disconnect the set of landing pages. However, in our case, this strategy was efficient almost exclusively for the full disruption. As we were not able to exhaustively compute the min-cut for all permutation of landing page samples, we are not able to leverage the theoretical performance of the min-cut method.

The choice between the Keyplayer and the min-cut methods will depend on the technical means available. With significant resources to commit to a disruption operation, it will be reasonable to opt for the min-cut method. Thus, by scanning all the sets of landing pages, it would be possible to find the best set of nodes to remove. On the contrary, the Keyplayer method is more appropriate in a resource constrained environment. It may not give the best set of nodes to remove but, for a full disruption, it gave results close to the min-cut method and gave better results for a partial disruption. As such, we chose to use the Keyplayer method as it was the most suitable for our problems and for our technical means.

4.6.5 Keyplayer detailed analysis

After evaluating different metrics to target nodes in a click-fraud ecosystem, we selected the Keyplayer method.

When looking at the results, we can see that removing the first three nodes has a large influence on the connectivity of the graph. At 4 nodes removed, the normalized fragmentation is above 80%, a high level of fragmentation. Moreover, it seems that removing more than 3-5 nodes has diminishing returns as the fragmentation plateaus. The high fragmentation obtained by removing a small number of actors is encouraging as this implies disruption could be relatively easily achieved.

The first three nodes selected for removal are *AdKernel*, *Deximedia* and *Vertamedia*. Once these three nodes are removed, the network becomes much more fragmented. The resulting graph is shown in Figure 4.7. By looking at the graph, it is clear that the removal of these three nodes does not isolate the Boaxxe root from its immediate neighborhood of radius 1. However, it does somewhat isolate that neighborhood from many legitimate ad networks and landing pages as more than 50% of them were disconnected from the search engine. Consequently, it becomes far more complex for Boaxxe to monetize their traffic.

However, we also need to make sure that it would not be possible for cyber criminals to quickly and efficiently replace these nodes. Therefore, we need to further analyze the nature of these three nodes removed by the algorithm.

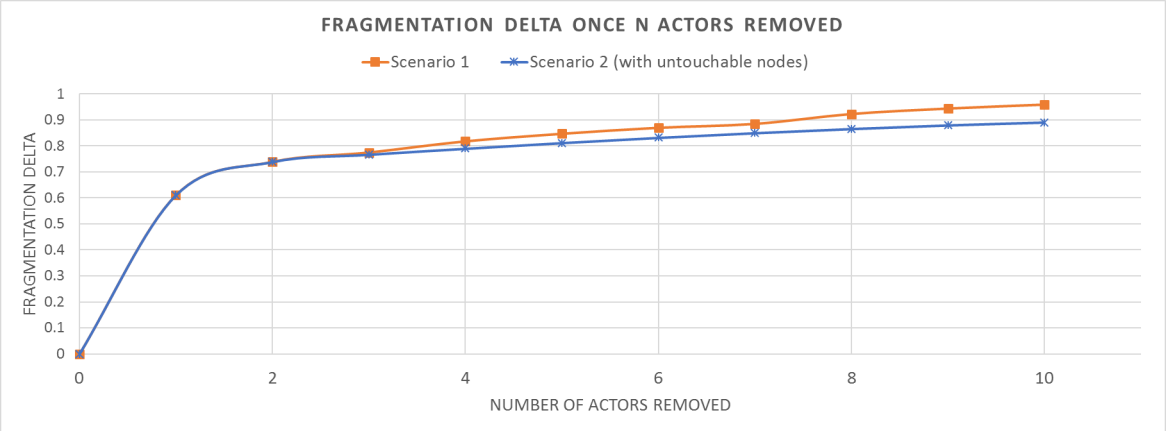


Figure 4.11 Impact of removal of actors on the fragmentation of the actor graph as measured by the difference in normalized fragmentation delta (y-axis) once a given number of actors (x-axis) are removed. In Scenario 1 any node can be removed, while in Scenario 2 nodes in the immediate neighborhood of the Boaxxe doorway SE are *untouchable* and cannot be removed.

Table 4.3 List of AdKernel customers

bluemediappc.com	terappc.com	dsmedianet.com	eliteppc.net	olmeramarketing.com
adsparkmedia.net	vokut.com	infinitywidget.com	finditquick.com	anytheengmedia.com
searcharbor.com	castramedia.com	marsfeeds.com	primusad.com	maxppc.com
vertoz.com	mindad-xml.com	mediacpc.com	ybrant-search.com	readywind.com
dogtownads.com	seodollars.com	**vespymedia.com**	madeofmedia.com	trafficaim.com
ctrtraffic.com	visitorsblitz.com	zipzipsearch.com	cpc-ads.com	globalsearchmedia.com
resultscpv.com	adconfide.com	xmladsystem.com	infinity-info.com	cubxml.com

AdKernel. This is an ad network Solution-as-a-Service (SaaS) provider. They provide all the infrastructure required to run an ad network. They are not necessarily directly involved in click-fraud, but their service is used by many suspicious ad networks. Table 4.3 summarizes the ad networks found in our Boaxxe dataset that use AdKernel services, with customers found in the Alexa Top 100,000 ranking in bold. Most of the ad networks present in the Alexa Top 100,000 ranking should be reputable companies. However, the highest Alexa-ranked ad network, *Vertoz* in this list, is also known for being involved in malvertising (malekalmorte, 2014). Moreover, when looking at the Virus Total Passive DNS database, we noticed that a number of additional ad networks with suspicious practices not present in our dataset also use AdKernel services.

Deximedia. There is limited information available on the web about this US media company except for a job offer in New York City and a discussion topic on the *blackhatworld* forum (Blackhatworld.com, 2014). This anecdotal evidence would seem to imply that Deximedia is not an ad network with a strong policy against click fraud. As such, it is difficult to know how much effort would be required by the underground to replace this node if it were removed from their ecosystem.

Vertamedia. The headquarters of this media company are located in New York City, but the two co-founders appear to work from Ukraine. According to its web site, this company has 20 employees and seems to participate in different digital media events. This suggests that they have a well-established position in the legitimate advertising market despite the fact that they are directly connected to the Boaxxe doorway SE. However, one of their domains, `c.feed-xml.com`, was embedded in the strings of several malware samples, including the well-known click-fraud malware Poweliks (Virustotal, 2016) and Bedep, another malware with a click-fraud payload (Frankoff, 2015).

The capacity to affect actors that are directly involved with Boaxxe or otherwise involved in illicit activities might be limited. Furthermore, targeting those actors might be inefficient as they can easily be replaced by fraudsters. For example, if we consider removing more than 2 nodes, we encounter some of the ad networks (Vertamedia) that are directly linked to the Boaxxe root. These are usually not attractive targets for disruption because of their fly-by-night nature.

For this reason, we considered an alternate application of the keyplayer technique where certain nodes were "protected" from the KPP-neg solver; we call these the *untouchable actors*. We ran the KPP-Neg solver again with all actors in the immediate neighborhood of the Boaxxe root labeled as untouchable. This enabled us to identify intermediary targets that are at a distance greater than one from Boaxxe.

Once this is done, it is interesting to note that some well-known ad networks appear in the list of nodes whose removal produces a high fragmentation delta, notably `advertise.com`, *eZanga* and *BlueLink Marketing*. `advertise.com` was already singled out for accepting click-fraud traffic by Bloomberg in the previously cited article in the section 4.2.2. In the same manner, eZanga and BlueLink Marketing were already suspected to buy botnet traffic in 2013, as detailed in an Adweek interview of Web fraud expert Ben Edelman (AdWeek, 2013). In summary, these ad networks appear to have recurring issues with botnet traffic.

Finally, the difference between the two curves is below 0.1 point. It means this second scenario is almost as efficient in term of disruption as the first scenario. They would thus seem to be attractive choices for disruption, as their positions in the legitimate ad market would make them ideal targets for pressure from their legitimate clients.

4.7 Discussion

In the last section, we presented disruption possibilities on the Boaxxe ecosystem. In this section, we will first discuss these results and secondly give insights on how to target the selected actors in practice.

4.7.1 Interpretation of results

At first glance, the most efficient way to stop fraudulent activity from a botnet would be to perform a take-down operation. However, take-downs are resource consuming and offer limited guarantees for long-term effectiveness. As an example, the ZeroAccess botnet was taken down in 2013 and resurrected at the beginning of 2015 (Stockley, 2015). As long as monetization schemes allow botnet operators to generate profits, they will continue to reinvest a good portion of them to rebuild any botnets taken down. In that light, disrupting the monetization scheme appears to be the best way to achieve long term results.

In the Boaxxe case, it is evident that the doorway search engines are the most critical nodes because they represent the root of the redirection tree. However, as seen in our dataset, the Boaxxe operator(s) already routinely change(s) the search engine domains. If the search engines are simply taken down, it would require very little effort to setup substitutes. That is not the case for the other economic actors involved in the Boaxxe click-fraud ecosystem. Even the fly-by-night advertisement companies that provide syndication for the Boaxxe search engines require incorporation and a veneer of legitimacy. Without those characteristics, it would not be possible to inject the click-fraud traffic in the legitimate market, where the victims reside.

In the previous section, we presented a comparison of techniques to select targets to disrupt Boaxxe's click-fraud ecosystem. It turns out that the Keyplayer technique is the most suitable for this type of study. Our results show that by removing only three carefully selected nodes we could impede the majority of click-fraud traffic to reach its victim. Indeed, we found that more than 50% of the landing pages would be disconnected from the doorway SE. This small number of targeted actors would seem to imply that the resources required for a successful disruption operation could be much smaller than the resources required to perform a traditional botnet take-down. If more resources were available, a crippling disruption of the ecosystem could be achieved with less than ten targeted actors. Furthermore, even if the actors that are closer to the Boaxxe root cannot be targeted (i.e. are "untouchable"), for example because they reside in unfriendly jurisdictions, it is still possible to obtain a significant disruption by targeting accessible actors. After all, as seen in our results, the difference in fragmentation delta between the scenarios with and without untouchable nodes is relatively small.

4.7.2 Targeting actors in practice

The keyplayer analysis technique allows us to identify the best candidates for disruption in the graph. However, it provides no guidance in how to "remove" the corresponding actors from the graph. Several options can be considered. First, legal action could be taken. While click-fraud may not be explicitly illegal in the criminal sense, there are no doubts that it represents a breach of contract in many cases.

Second, some of the ad networks that we identified in our dataset have well-known customers that could apply pressure on their providers. These ad networks enable cybercrime, whether willingly or by virtue of their negligence. They contribute to the success of criminals and undermine the digital ecosystem. For instance, as shown in our data, the Huffington Post, a well-known publisher, receives traffic from Deximedia. If customers such as the Huffington Post demanded stricter action against click-fraud by their traffic providers, these ad networks would be required to comply, or else lose the business from these customers. To do so, advertisers should also change their practices to assess the quality of the traffic they receive. In particular, we advocate for the use of better metrics to measure the Return-on-Investment (ROI) on Internet publicity. The prevalent use of metrics such as volume of incoming traffic, page counts, etc. do not allow the advertisers to differentiate bot traffic from human traffic. Thus, it does not create an incentive, for the defrauded advertisers, to change for a better ad network. As a consequence, it is an indirect cause of this publicity fraud phenomenon. Furthermore, we also need to remember that ad networks frequently sell traffic to each other.

As such, other ad networks could threaten to ban bad apples from their ad exchanges or standing purchasing agreements in a similar manner as discussed above for advertisers.

Third, we showed that the targeting of AdKernel, an ad network SaaS provider, caused the most impact on the click-fraud ecosystem. This is not surprising as it drastically decreases the barriers to entry in the advertising market. With this kind of SaaS service, anyone can launch an ad network, or even relaunch it when its reputation becomes too poor. In this light, SaaS providers could be made to accept more accountability for the activities of their customers, and could offer help in removing any known bad actors that are abusing their services.

Finally, the advertising industry launched in 2014 the Trustworthy Accountability Group (Trustworthy Accountability Group, 2016). It aims to regulate the advertising market by giving a certification to companies that can be trusted. Moreover, they developed *Payment ID*, a system in which each click or impression is given a unique identifier. Thus, when an advertiser detects invalid traffic, he can follow the supply chain and blacklist the fraudulent traffic providers at the origin of the redirect chain.

4.8 Related work

While no other research group has specifically tackled the problem of disrupting the click-fraud ecosystem, a number of researchers have provided insight on the world of click fraud.

One of the first analysis of a click-fraud malware binary was that of Clickbot.A in 2007 (Daswani and Stoppelman, 2007). The authors detail the low-noise techniques used by the malware operator to perform click-fraud and present an estimation of the cost of the fraud for advertisers. While this malware did not cause any damage to the infected computer or its owner, the authors claim that ad networks, anti-virus companies, advertisers and publishers should work together to disrupt such activities. The rationale is that these activities generate a large amount of money for criminals and create incentives for them to cause harm to users. Later, Miler *et al.* (Miller et al., 2011) examined two different click-fraud malware, 7cy and Fiesta, in order to compare them with Clickbot.A. They found new techniques employed by this malware to mimic the behavior of a human browsing web sites in order to evade fraud detection. In 2014, Pearce *et al.* (Pearce et al., 2014) made a detailed analysis of the ZeroAccess click-fraud malware and its monetization strategy. However, the paper was limited to describing the ecosystem rather than to find ways to disrupt it. Thomas *et al.* focused on studying ad injections, a form of advertisement fraud involving extensions that modify the web page DOM (Thomas et al., 2015).

Alrwais *et al.* (Alrwais et al., 2012) studied the effect of the FBI's Operation Ghost Click. This was a large take-down of an ad-fraud botnet that was using rogue DNS server to hijack valid ads and replace them by ads supplied by the malware, a form of clickjacking. However, this operation did not disrupt the advertising ecosystem related to the malware. Chances are that parts of this ecosystem have been reused by other click-fraud malware since.

Other studies have focused on the ad ecosystem itself. Stone-Gross *et al.* (Stone-Gross et al., 2011) examined an ad exchange system to understand how these systems can be abused by criminals to generate profit. Zhang *et al.* (Zhang et al., 2011) bought traffic from different traffic providers for their own web site and evaluated, for each provider, the quality of the traffic. They found that the traffic coming from bulk providers was of poor quality in comparison to Google Adwords. Snyder *et al.* (Snyder and Kanich, 2015) studied affiliate marketing fraud. Dave *et al.* (Dave et al., 2012)(Dave et al., 2013) focused on how to detect fraudulent clicks by using appropriate metrics. Recently, Javed *et al.* (Javed et al., 2015) showed the existence of traffic exchange services that provide an alternate way of generating fraudulent clicks rather than using automated click bots.

4.9 Conclusion

In this paper, we described the click-fraud ecosystem of Boaxxe/Miuref, a well-known click-fraud botnet. We collected click-fraud network traces from self-infected Boaxxe bots in a 7-month longitudinal study. By reconstructing the redirection chains of the automated click-fraud activity, it was possible to adequately sample the actor graph of the ecosystem. We also collected click-fraud chains from the Ramdo doorway search engine and used this dataset to control our experiment. In particular, the Ramdo data set allowed to validate the correctness of our chain reconstruction algorithm.

We then compared different strategies to find the best set of critical nodes of the fraud ecosystem. We showed that the Keyplayer technique is the most suitable for our problem. The same result was also reproduced for the actor graph reconstructed from the Ramdo data set.

We found that, by removing a very limited number of actors, the monetizing capacity of the botnet could be seriously disrupted. Of these actors, one of the most interesting is AdKernel, a Solution-as-a-Service provider for ad networks. This is not surprising as it enables companies to enter the advertising market by reducing barriers to entry. This illustrates the importance of preventing the use of these services by criminals.

Finally, as click fraud and other types of ad-based monetizing schemes become an increasingly

important source of revenue for criminals, we argue that ecosystem disruption techniques based on information acquired from the analysis of redirection chains should be more widely used. While botnet take-downs can achieve short term success, they are less efficient in the long term.

An assumption of our work is that the redirection chains reconstructed from client network traces capture all of the business relationships involved in Internet advertisement. However, it would be surprising if there were blind trust between different online advertising actors, leading to invisible transactions. A way to validate this assumption would be by cross-referencing our data with that of actual ad networks, for example through industry-wide data sharing initiatives, such as those mentioned in Section 4.8.

Future work should widen our study. As seen before, an ad network directly linked to the Boaxxe search engines was also seen in Bedep traffic, another click-fraud malware. This suggests that it could be worthwhile to collect network traces from several click-fraud botnets and apply the same key player method and compare the results, in order to see whether the same disruption targets apply to several click-fraud botnets. Similarly, the method could be applied to other ad-based monetization schemes such as black search engine optimization and adware. This more global study might enable us to disrupt more generally the ad-based fraud ecosystem that is a threat to the web economy.

Acknowledgements

We thank Pierre-Marc Bureau for providing expertise on Ramdo, and for reviewing and providing insightful suggestions for improvement of this article.

CHAPITRE 5 DISCUSSION GÉNÉRALE

Dans le chapitre précédent, nous avons présenté la méthodologie de notre étude ainsi que les résultats obtenus. Dans ce chapitre, nous ferons le lien entre les objectifs énoncés à la section 1.3 avec le travail présenté dans l'article.

5.1 Modélisation de l'écosystème à partir de l'activité d'un logiciel malveillant de fraude au clic

Notre premier objectif est de tenter de reconstruire l'écosystème de fraude au clic à partir des informations récoltées d'un logiciel malveillant de fraude au clic. Cet objectif est constitué de plusieurs étapes dont la récolte de données, le traitement de celles-ci et enfin leur utilisation afin de modéliser un écosystème.

5.1.1 Données d'un logiciel malveillant de fraude au clic

La première étape de notre premier objectif de recherche consistait à trouver les informations pertinentes à extraire de l'activité d'un logiciel malveillant de fraude publicitaire.

Tel que vu à la section 2.1.1, les régies publicitaires font de l'arbitrage en temps réel. Ce processus est généralement répété plusieurs fois pour la même publicité. De plus, cet arbitrage est représenté d'un point de vue réseau par des redirections entre les différents revendeurs. Ainsi, il est possible de connaître l'ensemble des revendeurs à partir des traces réseaux d'un client. Nous avons donc choisi de collecter les données réseau d'un logiciel malveillant de fraude publicitaire, Boaxxe. Pour cela, nous avons infecté nous-mêmes une machine virtuelle et enregistré les traces réseaux dans un fichier *Packet CAPture* (PCAP).

5.1.2 Reconstruction des chaînes

Nous avons vu précédemment que les relations économiques entre les différents acteurs étaient modélisées par des redirections. Or, ces redirections n'apparaissent pas directement dans les fichiers réseaux étant donné que le protocole HTTP est sans état.

Ainsi, nous avons développé une méthode permettant de reconstruire les chaînes de redirections à partir d'un fichier réseau. Cette méthode a été validée grâce à des données issues du logiciel malveillant Ramdo et collectées grâce à un navigateur web instrumenté. Ces données sont considérées comme fiables puisque l'instrumentation dans le navigateur nous permet de

capturer chaque redirection, c'est-à-dire chaque changement d'adresse. Ainsi, il est possible d'avoir une liste exacte des URL successives de la chaîne de redirection.

Une fois les chaînes reconstruites, il est possible de construire un graphe montrant l'ensemble des liens entre les différents nœuds réseau.

5.1.3 Agrégation des nœuds

Les chaînes obtenues grâce à notre méthode de reconstruction permettent d'identifier les redirections entre les nœuds réseaux, à savoir des adresses IP et des noms de domaine. Or, une seule entité économique peut posséder plusieurs nœuds réseaux. Ainsi, le graphe obtenu précédemment n'est pas une représentation exacte de l'écosystème.

Afin de palier à ce problème, nous avons développé une méthodologie, utilisant des données de source ouverte, afin d'identifier des groupes de nœuds réseaux appartenant au même acteur. Ces données comprennent des données de Whois, de DNS passif, de certificats *Secure Sockets Layer* (SSL) ou encore des *tracking code* tels que des codes Google Analytics ou Google Adsense.

Une fois les différents indices récoltés, nous avons procédé à une validation manuelle de chaque fusion de nœud. En effet, il est par exemple possible que différents acteurs utilisent le même service d'hébergement partagé (cloud) et aient donc la même adresse IP.

Finalement, une fois les nœuds appartenant aux mêmes acteurs fusionnés, nous avons une représentation de l'écosystème.

5.1.4 Validité de la première question de recherche

Nous avons récolté des données d'un logiciel malveillant de fraude au clic, Boaxxe, et traité celles-ci afin qu'elles soient exploitables pour l'analyse. Nous avons aussi vérifié la validité de ce traitement grâce à d'autres données, issues du logiciel malveillant Ramdo. Enfin, nous avons pu reconstruire un graphe des relations économiques des acteurs par lesquels circule le trafic issu de Boaxxe. Ce graphe est donc une représentation de l'écosystème de la fraude associée à Boaxxe. Ces différentes étapes montrent qu'il est possible de modéliser un écosystème en observant l'activité d'un logiciel malveillant de fraude au clic. Ceci répond donc à notre première question de recherche.

5.2 Techniques de perturbation

Notre objectif de recherche est non seulement d'obtenir un écosystème de fraude au clic mais aussi de trouver des manières de le perturber. Ainsi, il s'est révélé nécessaire de comparer différentes techniques permettant de cibler des acteurs avant d'évaluer les résultats de cette perturbation.

5.2.1 Comparaison de différentes techniques

Dans le cas de la fraude au clic, l'objectif de la perturbation de l'écosystème est d'empêcher la monétisation du trafic provenant du moteur de recherche d'entrée du logiciel malveillant de fraude publicitaire. Cela consiste donc à empêcher le transit du trafic de ce moteur de recherche vers les sites des annonceurs. Pour cela, il est nécessaire de couper tous les liens qui permettent au trafic malveillant d'atteindre les annonceurs. Autrement dit, nous souhaitons disjoindre le moteur de recherche des sites des annonceurs.

Afin d'identifier les acteurs les plus importants d'un écosystème, c'est-à-dire d'un réseau (graphe), il est possible d'utiliser différentes métriques d'analyse de réseaux sociaux. Parmi celles-ci, on trouve les métriques de centralité tel que la centralité de degré, la centralité de proximité ou la centralité d'intermédiarité. Par ailleurs, nous avons aussi inclus la méthode du Keyplayer, qui est une méthode utilisée en criminologie afin de perturber des réseaux criminels. Enfin, nous avons aussi comparé ces méthodes à la méthode de la coupe minimum qui permet de trouver les nœuds à éliminer afin de disjoindre deux nœuds du graphe.

Afin de valider nos résultats, nous avons effectué la même comparaison sur le jeu de données issu du logiciel malveillant Ramdo. Les résultats obtenus sont identiques à ceux obtenus sur Boaxxe.

Cette comparaison nous a notamment permis d'identifier la technique la plus appropriée à notre problème, celle du Keyplayer.

5.2.2 Résultats de la technique Keyplayer

En appliquant la technique du Keyplayer à l'écosystème de Boaxxe, nous avons réussi à le perturber efficacement. En effet, la suppression de trois nœuds de régie publicitaire permet d'empêcher le moteur de recherche de Boaxxe d'atteindre plus de 50% des annonceurs. De plus, la suppression de 10% des nœuds (12 nœuds) permet de rendre hors d'atteinte près de 90% des annonceurs.

Ainsi, les acteurs ciblés grâce à la technique du Keyplayer semblent avoir une grande impor-

tance au sein de l'écosystème. En effet, supprimer une faible proportion de nœuds permet d'obtenir une grande perturbation de l'écosystème.

Enfin, nous avons présenté les trois premiers acteurs à cibler. Parmi ceux-ci, le plus intéressant est Adkernel qui est un acteur légitime. Cette entreprise propose un service de régie publicitaire en tant que service qui est utilisée par de nombreux fraudeurs. Ainsi, il semble que des entreprises servent de support à la fraude au clic.

L'écosystème de la fraude est donc constitué de fraudeurs mais aussi d'entreprises qui sont abusées et servent de support à la fraude. Celles-ci sont donc victimes de part leur ignorance ou leur manque de contrôle sur les produits ou services mis à la disposition de leurs clients. Il pourrait donc être intéressant de les sensibiliser et font sûrement partie de la solution au problème de la fraude au clic.

5.2.3 Mise en pratique

Nos expérimentations de perturbation ont été réalisées d'une manière théorique sur le graphe. Or, il n'est pas possible en pratique de supprimer un acteur. Cela consisterait donc, comme évoqué précédemment, à faire pression sur cet acteur afin qu'il arrête d'être un point de passage du trafic lié à la fraude au clic.

Par ailleurs, nous avons préconisé aux annonceurs d'être le plus vigilant possible quant à la qualité du trafic qu'ils reçoivent puisqu'ils sont les premiers à pouvoir faire pression sur leurs régies publicitaires.

Ainsi, la comparaison de différentes techniques de perturbation et l'analyse détaillée des résultats du Keyplayer répondent à la deuxième question spécifique de recherche.

5.3 Atteinte de l'objectif de recherche et limitations

Dans ce travail, nous avons récolté des données d'un logiciel malveillant de fraude au clic, Boaxxe. Nous avons ensuite reconstruit l'écosystème de la fraude au clic de Boaxxe grâce à ces données puis nous y avons évalué différentes possibilités de perturbation.

La solution proposée dans ce travail présente trois principales limitations. Tout d'abord, nous avons fait l'hypothèse que toutes les transactions entre les différents acteurs, pour les publicités que nous avons étudiées, apparaissaient dans les chaînes de redirection. Cependant, il est possible qu'il existe des accords entre régies publicitaires qui n'apparaissent pas dans ces chaînes. Or, afin de pouvoir quantifier le trafic, une régie publicitaire doit soit avoir une totale confiance dans les régies publicitaires présentes dans la chaîne ou soit faire transiter le

trafic par sa propre infrastructure. Cette hypothèse est donc vraisemblable puisque, d'après nos connaissances du marché de la publicité en ligne, il n'y a que très peu de confiance entre les acteurs, obligeant ceux-ci à faire transiter le trafic par leurs propres serveurs.

Ensuite, nous n'avons pas tenu compte des poids dans les relations entre les acteurs de l'écosystème, c'est-à-dire du nombre de fois qu'un arc du graphe apparaît dans notre jeu de données. En effet, ce poids mesuré entre deux acteurs est un échantillonnage du vrai volume de transaction lié à Boaxxe entre ces deux acteurs de l'écosystème. Or, nous n'avons pas d'indications quant à sa représentativité statistique. Ainsi, nous avons décidé de ne pas en tenir compte afin d'éliminer ce biais. Par ailleurs, le vrai volume de transaction lié à Boaxxe est lui-aussi un échantillonnage du volume de transaction total entre ces deux acteurs. Ainsi, nous ne pouvons pas déterminer l'influence véritable des acteurs les uns par rapport aux autres. Cependant, l'objectif est de couper toutes les relations entre le moteur de recherche d'entrée et les annonceurs et n'est donc pas limité par ce manque d'information sur le poids des arcs.

Enfin, notre étude longitudinale n'a porté que sur un seul logiciel malveillant. Nos conclusions sont donc limitées à l'écosystème de la fraude au clic de Boaxxe. Cependant, la méthodologie employée n'est pas spécifique à ce logiciel malveillant et pourrait être facilement réutilisée. On notera que les données de Ramdo, utilisées ici seulement à des fins de vérification de la méthodologie de reconstruction des chaînes et de la comparaison des méthodes de perturbation, présentent plusieurs acteurs communs avec l'écosystème de Boaxxe. Plus précisément, 28% des acteurs de l'écosystème de Boaxxe sont présents dans l'écosystème de Ramdo. Cela renforce donc l'intérêt d'étendre l'étude à d'autres logiciels malveillants de fraude au clic.

CHAPITRE 6 CONCLUSION

Dans la section précédente, nous avons discuté nos résultats et nous les avons comparés avec nos objectifs de recherche. Dans cette section, nous commencerons par une synthèse des travaux puis nous reviendrons sur les limitations de nos travaux. Enfin, nous détaillerons les améliorations qui pourraient y être apportées.

6.1 Synthèse des travaux

Au cours de ce travail, nous avons souhaité mieux connaître le fraude au clic afin d'identifier des méthodes permettant de réduire ce phénomène.

Tel que vu au chapitre 2, les recherches précédentes se sont concentrées à l'analyse des logiciels malveillants de fraude au clic et aux possibilités de démantèlement de ceux-ci. Cependant, ceci ne fait que ralentir les fraudeurs en les obligeant à développer de nouveaux programmes malveillants toujours plus perfectionnés. D'autre part, des recherches ont aussi proposé des méthodes permettant une détection plus efficace des clics frauduleux. Or, cela déclenche aussi une course à l'armement avec les fraudeurs qui chercheront à passer à travers ces filtres en créant des robots imitant le comportement humain. Ainsi, si les recherches précédentes ont sûrement permis de freiner le phénomène, elles n'ont pas coupé l'incitatif économique à frauder.

Dans la section 1.2.3, nous avons fait l'analogie entre la fraude au clic et les fausses pharmacies sur internet. En effet, l'écosystème de cette fraude a été perturbé en étudiant la chaîne de valeurs entre les différents acteurs. Il a ainsi été possible d'identifier des points critiques qui sont les processeurs de paiement de carte de crédit. Ainsi, en coupant les liens avec ces processeurs de paiement peu scrupuleux, il a été possible de ralentir le flot d'argent vers les fraudeurs. Étant donné que la fraude au clic passe par un grand nombre d'intermédiaires, nous avons évoqué l'idée d'étudier la chaîne de valeur de la fraude au clic. Ceci permettrait d'une part de cartographier l'écosystème associé et d'autre part d'identifier les points critiques.

Cependant, nous n'avons pas directement accès aux transactions financières constituant la chaîne de valeur. Nous avons donc utilisé les chaînes de redirection publicitaire comme représentation de cette chaîne de valeur. En effet, les achats et reventes de publicité sont représentées, du côté de l'utilisateur, par une redirection de l'agent utilisateur.

La première étape de nos travaux a donc consisté à collecter des chaînes de redirection générées par un logiciel malveillant de fraude au clic, Boaxxe. Étant donné que ces données

étaient des traces réseaux brutes, nous avons développé une méthodologie permettant de reconstruire les chaînes de redirection à partir de fichiers PCAP. Nous avons aussi ajouté une étape d'agrégation des nœuds réseaux, basée sur des données à source ouverte, afin d'identifier les adresses IP et noms de domaine appartenant à une même organisation. Une fois cette étape effectuée, nous avons accès à une représentation de l'écosystème de la fraude au clic liée à Boaxxe.

La seconde étape de nos travaux a consisté à évaluer différentes stratégies de perturbation sur cet écosystème. Il s'est avéré que seulement deux des méthodes évaluées sont adaptées à notre problème : le Keyplayer et la coupe minimum. Cependant, la coupe minimum nécessite des moyens bien plus conséquents. Nous avons donc choisit d'utiliser le Keyplayer.

La dernière étape de notre travail a été d'analyser les résultats obtenu grâce à la technique du Keyplayer. Nous avons identifié les acteurs les plus importants de l'écosystème. Parmi ceux-ci, on notera la présence d'Adkernel qui est une régie publicitaire en tant que service, permettant à des personnes sans connaissance technique d'opérer une régie publicitaire.

Ainsi, nous avons reconstruit l'écosystème de la fraude au clic de Boaxxe et nous avons proposé des solutions pour le perturber. Ceci montre qu'il est possible d'avoir une approche différente de celles actuellement utilisées afin de diminuer très fortement l'incitatif économique à faire de la fraude au clic.

6.2 Limitations de la solution proposée

Tel qu'exposé dans la section 5.3, les principales limitations de notre recherche sont l'hypothèse que toutes les transactions entre les acteurs apparaissent dans les chaînes de redirection, l'absence de prise en compte des poids des arcs, c'est-à-dire du nombre de fois qu'un arc du graphe apparaît dans le jeu de données, et l'étude limitée à un seul logiciel malveillant, Boaxxe.

Cependant, l'hypothèse est très vraisemblable, étant donné le peu de confiance entre les différents acteurs du monde de la publicité en ligne et l'absence de poids permet quand même d'identifier tous les chemins permettant le transit du trafic malveillant. Enfin, la méthodologie employée pourra être facilement réutilisée sur un autre logiciel malveillant de fraude publicitaire.

6.3 Contributions

Notre recherche a contribué à une meilleure connaissance de la fraude au clic. En particulier, nous avons montré qu'il était possible d'avoir une représentation de l'écosystème de la fraude par l'étude dynamique d'un logiciel malveillant.

De plus, nous avons aussi montré qu'il était possible de perturber cet écosystème grâce à des stratégies bien choisies telles que le Keyplayer. Ceci permet donc de comprendre comment diminuer l'incitatif économique que constitue le marché de la publicité en ligne.

6.4 Améliorations futures

Les travaux futurs visent d'une part à répondre aux limitations énoncées précédemment et d'autre part à étendre notre recherche.

Afin de vérifier la validité de notre hypothèse de base, soit la présence de toutes les transactions entre les acteurs dans les chaînes de redirection, il serait possible de croiser les données récoltées à partir d'un poste client infecté aux données d'une régie publicitaire partenaire. Ainsi, cela permettrait de vérifier la présence ou l'absence d'intermédiaires cachés.

Ensuite, pour pouvoir prendre en compte les poids des arcs entre les différents acteurs, il serait nécessaire d'avoir un échantillon raisonnable des transactions entre ces acteurs. D'une part, une possibilité est d'avoir une vue globale du trafic issu de toutes les machines infectées par Boaxxe, ou au moins, un échantillon représentatif des machines infectées. Cependant, cela demande aussi de pouvoir connaître la taille du réseau de machines zombies avec précision pour déterminer la taille de l'échantillon. D'autre part, l'accès à des données d'acteurs intermédiaires de l'écosystème, comme le volume de transaction, pourrait aussi permettre de valider l'échantillonnage des transactions et donc le poids des arcs. La connaissance de ces poids nous permettrait de pouvoir déterminer la force des relations et donc d'en tenir compte pour faire des scénarios de perturbation partielle de l'écosystème de la fraude au clic. Ainsi, une collaboration avec des acteurs intermédiaires motivés à lutter contre la fraude publicitaire serait souhaitable et permettrait d'utiliser à plus grande échelle les méthodes développées ici.

Enfin, la même méthodologie pourrait être facilement utilisée à l'étude d'autres logiciels malveillants de fraude au clic. Ceci permettra ensuite une comparaison entre les différents écosystèmes afin d'identifier de potentiels acteurs communs. De plus, il sera possible d'évaluer les conséquences de la perturbation d'un écosystème sur un autre écosystème.

D'autre part, notre recherche pourrait être étendu à des domaines adjacents de la fraude au

clic.

Tout d'abord, il pourrait être intéressant d'étudier les logiciels potentiellement indésirables (*adware*), qui ont souvent recours à la publicité. La même méthodologie pourrait être employée afin de cartographier l'écosystème lié. Il serait ensuite possible d'effectuer des comparaisons avec l'écosystème de la fraude au clic afin d'identifier de potentiels acteurs communs.

Ensuite, récolter des données sur la publicité malveillante (*malvertising*) permettrait aussi de cartographier les différents acteurs de la publicité en ligne par lesquels passent ce contenu malveillant. En utilisant les mêmes techniques de perturbation, il deviendrait possible de couper les chemins entre les sites des éditeurs et les sites des annonceurs malveillants, qui utilisent par exemple des exploit-kit afin d'installer automatiquement des logiciels malveillants à l'insu de l'utilisateur.

Enfin, nous n'avons pas étudié les sites des annonceurs vers lesquels sont redirigé le trafic malveillant généré par Boaxxe. Or, un grand nombre d'entre eux semblent être des sites uniquement destinés à générer de l'argent grâce à la publicité en ligne. Il pourrait donc être intéressant de classifier les différents sites des annonceurs afin de pouvoir déterminer quels sont ceux ayant le plus grand intérêt à recevoir des visiteurs réellement humains. Par exemple, un site de commerce en ligne gagnera difficilement de l'argent avec du trafic automatisé quand un blog pourra bénéficier des faiblesses des contrôles anti-fraude pour générer de l'argent grâce à des publicités vues par des robots. Cette étude est particulièrement importante dans l'objectif de sensibilisation des annonceurs à la fraude au clic puisque ce sont notamment eux qui doivent prendre des mesures afin de couper leurs relations commerciales avec les intermédiaires les moins scrupuleux.

RÉFÉRENCES

"2015 full year - digital advertising revenue report", April 2016. En ligne : http://www.iab.com/wp-content/uploads/2016/04/IAB-Internet-Advertising-Revenue-Report-FY-2015.pdf

AdWeek, "The six companies fueling an online ad crisis", 2013. En ligne : http://www.adweek.com/news/advertising-branding/six-companies-fueling-online-ad-crisis-150160

S. Alrwais, K. Yuan, E. Alowaisheq, Z. Li, et X. Wang, "Understanding the dark side of domain parking", dans *Proceedings of the 23rd USENIX Conference on Security Symposium*, série SEC'14. Berkeley, CA, USA : USENIX Association, 2014, pp. 207–222.

S. A. Alrwais, A. Gerber, C. W. Dunn, O. Spatscheck, M. Gupta, et E. Osterweil, "Dissecting ghost clicks : Ad fraud via misdirected human clicks", dans *Proceedings of the 28th Annual Computer Security Applications Conference*, série ACSAC '12. New York, NY, USA : ACM, 2012, pp. 21–30.

ANA - White OPS, "The Bot Baseline - Fraud in Digital Advertising", Rapp. tech., 2014. En ligne : https://www.ana.net/getfile/21853

——, "The Bot Baseline - Fraud in Digital Advertising", Rapp. tech., 2015. En ligne : http://www.ana.net/getfile/23332

K. Asdemir, O. Yurtseven, et M. A. Yahya, "An Economic Model of Click Fraud in Publisher Networks", *International Journal of Electronic Commerce*, vol. 13, no. 2, pp. 61–90, Déc. 2008.

T. Baysinger, "The online industry is losing $8 billion a year, and ad blocking is the least of its worries", 2015. En ligne : http://www.adweek.com/news/advertising-branding/how-online-industry-losing-8-billion-every-year-168389

T. Berners-Lee, "The HTTP protocol as implemented in W3C", 1991. En ligne : https://www.w3.org/Protocols/HTTP/AsImplemented.html

Blackhatworld.com, 2015. En ligne : http://www.blackhatworld.com/blackhat-seo/other-ppc-networks/759368-we-looking-new-traffic-sources.html

——, “What is deximedia.com ?” 2014. En ligne : http://www.blackhatworld.com/blackhat-seo/facebook/659769-what-deximedia-com.html

S. P. Borgatti, “Identifying sets of key players in a social network”, *Computational and Mathematical Organization Theory*, vol. 12, no. 1, pp. 21–34, Avr. 2006.

S. Borgatti, “Keyplayer program”. En ligne : http://www.analytictech.com/keyplayer/keyplayer.htm

J. Calvet, “Boaxxe adware : ‘a good ad sells the product without drawing attention to itself’ pt 1”, 2014. En ligne : http://www.welivesecurity.com/2014/01/14/boaxxe-adware-a-good-ad-sells-the-product-without-drawing-attention-to-itself-pt-1/

J. Calvet, C. R. Davis, J. M. Fernandez, J.-Y. Marion, P.-L. St-Onge, W. Guizani, P.-M. Bureau, et A. Somayaji, “The case for in-the-lab botnet experimentation : creating and taking down a 3000-node botnet”, dans *Proc. 26th Annual Computer Security Applications Conf. (ACSAC)*. ACM, Dec 2010, pp. 141–150.

R. Clayton, T. Moore, et N. Christin, “Concentrating Correctly on Cybercrime Concentration”, dans *Proceedings of the Fourteenth Workshop on the Economics of Information Security (WEIS)*, Delft, Netherland, 2015.

Counter Threat Unit (CTU) Research Team, “The untold story of the ramdo click-fraud malware”, 04 2016. En ligne : https://www.secureworks.com/blog/ramdo-click-fraud-malware

M. E. Crovella et A. Bestavros, “Self-similarity in world wide web traffic : evidence and possible causes”, *Networking, IEEE/ACM Transactions on*, vol. 5, no. 6, pp. 835–846, 1997.

N. Daswani et M. Stoppelman, “The anatomy of Clickbot.A”, dans *Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets*, série HotBots’07. Berkeley, CA, USA : USENIX Association, 2007, pp. 11–11.

V. Dave, S. Guha, et Y. Zhang, “Measuring and fingerprinting click-spam in ad networks”, *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 175–186, 2012.

——, “Viceroi : Catching click-spam in search ad networks”, dans *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, série CCS ’13. New York, NY, USA : ACM, 2013, pp. 765–776.

C. R. Davis, J. M. Fernandez, S. Neville, et J. McHugh, “Sybil attacks as a mitigation strategy against the storm botnet”, dans *Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on.* IEEE, 2008, pp. 32–40.

C. R. Davis, S. Neville, J. M. Fernandez, J.-M. Robert, et J. Mchugh, “Structured peer-to-peer overlay networks : Ideal botnets command and control infrastructures ?” dans *European Symposium on Research in Computer Security.* Springer Berlin Heidelberg, 2008, pp. 461–480.

L. Dritsoula et J. Musacchio, “A game of clicks : economic incentives to fight click fraud in ad networks”, *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 4, pp. 12–15, 2014.

D. Décary-Hétu et B. Dupont, “The social network of hackers”, *Global Crime*, vol. 13, no. 3, pp. 160–175, Août 2012.

B. Elgin, M. Riley, D. Kocieniewski, et J. Brustein, “The fake traffic schemes that are rotting the internet”, 2015. En ligne : `http://www.bloomberg.com/features/2015-click-fraud/`

S. F. Everton, *Disrupting dark networks.* Cambridge University Press, 2012, vol. 34.

M. Faou, J. Calvet, P.-M. Bureau, A. Lemay, et J. M. Fernandez, “The good, the bad & the ugly : The advertiser, the bot & the traffic broker”, dans *Virus Bulletin Conference*, Oct 2016.

M. Faou, A. Lemay, D. Décary-Hétu, J. Calvet, F. Labrèche, M. Jean, B. Dupont, et J. M. Fernandez, “Follow the traffic : stopping click fraud by disrupting the value chain”, Ecole Polytechnique de Montréal, Rapp. tech., 2016.

R. Fielding, J. Gettys, J. Mogul, H. Frystyk, et T. Berners-Lee, “RFC 2068 : Hypertext Transfer Protocol – HTTP/1.1”, 1997. En ligne : `https://www.ietf.org/rfc/rfc2068.txt`

R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, et T. Berners-Lee, “RFC 2616 : Hypertext Transfer Protocol – HTTP/1.1”, 1999. En ligne : `https://www.ietf.org/rfc/rfc2616.txt`

S. Frankoff, “Bedep ad-fraud botnet analysis – exposing the mechanics behind 153.6m defrauded ad impressions a day”, 05 2015. En ligne : `https://sentrant.com/2015/05/20/bedep-ad-fraud-botnet-analysis-exposing-the-mechanics-behind-153-6m-defrauded-ad-impressions-a-day/`

P. Gill, V. Erramilli, A. Chaintreau, B. Krishnamurthy, K. Papagiannaki, et P. Rodriguez, “Follow the money : Understanding economics of online aggregation and advertising”, dans *Proceedings of the 2013 Conference on Internet Measurement Conference*, série IMC ’13. New York, NY, USA : ACM, 2013, pp. 141–148.

P. H. C. Guerra, D. Guedes, J. Wagner Meira, C. Hoepers, M. Chaves, et K. Steding-Jessen, “Exploring the spam arms race to characterize spam evolution”, dans *Proceedings of the 7th Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS), Redmond, WA*, 2010.

Hewlett Packard Enterprise, *The Business of Hacking*, 2016. En ligne : `http://static.politico.com/b9/55/4e3ce4cc41d88401e264dcacc35c/hpe-security-research-business-of-hacking-may-2016.pdf`

A. Hidayat, “Phantomjs”, 2016. En ligne : `http://phantomjs.org`

M. Javed, C. Herley, M. Peinado, et V. Paxson, “Measurement and analysis of traffic exchange services”, dans *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, série IMC ’15. New York, NY, USA : ACM, 2015, pp. 1–12.

P. Kalnay et J. Horejsi, “Notes on click-fraud : American story”, dans *Virus bulletin conference*, 2014, pp. 118–129.

T. Krazit, “Yahoo settles pay-per-click fraud suit”, 2009. En ligne : `http://www.cnet.com/news/yahoo-settles-pay-per-click-fraud-suit/`

B. Krebs, “Rogue pharma, fake av vendors feel credit card crunch”, 2012. En ligne : `http://krebsonsecurity.com/2012/10/rogue-pharma-fake-av-vendors-feel-credit-card-crunch/`

——, “Chronopay’s scareware diaries”, 2011. En ligne : `http://krebsonsecurity.com/2011/03/chronopays-scareware-diaries/`

malekalmorte, “directrev malvertising lead to Zbot | malekal’s site”, Jan. 2014. En ligne : `http://www.malekal.com/directrev-malvertising-lead-to-zbot/`

R. McCormick, “AT&T reportedly playing dirty tricks to serve extra ads through airport hotspot”, 2015. En ligne : `http://www.theverge.com/2015/8/25/9208919/at-t-wifi-hotspots-insert-extra-ads-traffic`

H. Mekky, R. Torres, Z.-L. Zhang, S. Saha, et A. Nucci, “Detecting malicious HTTP redirections using trees of user browsing activity”, dans *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 1159–1167.

B. Miller, P. Pearce, C. Grier, C. Kreibich, et V. Paxson, “What’s clicking what ? techniques and innovations of today’s clickbots”, dans *Proceedings of the 8th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, série DIMVA’11. Berlin, Heidelberg : Springer-Verlag, 2011, pp. 164–183.

B. Mungamuru, S. Weis, et H. Garcia-Molina, “Should ad networks bother fighting click fraud ?(yes, they should.)”, 2008. En ligne : `http://ilpubs.stanford.edu:8090/840/`

C. Neasbitt, R. Perdisci, K. Li, et T. Nelms, “Clickminer : Towards forensic reconstruction of user-browser interactions from network traces”, dans *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, série CCS ’14. New York, NY, USA : ACM, 2014, pp. 1244–1255.

P. Pearce, V. Dave, C. Grier, K. Levchenko, S. Guha, D. McCoy, V. Paxson, S. Savage, et G. M. Voelker, “Characterizing large-scale click fraud in ZeroAccess”, dans *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, série CCS ’14. New York, NY, USA : ACM, 2014, pp. 141–152.

J. Seitz, “Automatically Discover Website Connections Through Tracking Codes | Automating OSINT Blog”, Août 2015. En ligne : `http://www.automatingosint.com/blog/2015/08/osint-discover-shared-tracking-code-between-domains/`

D. Silverman, “IAB internet advertising revenue report”, 2016. En ligne : `http://www.iab.com/wp-content/uploads/2016/04/IAB-Internet-Advertising-Revenue-Report-FY-2015.pdf`

P. Snyder et C. Kanich, “No please, after you : Detecting fraud in affiliate marketing networks”, dans *Proceedings of the Workshop on the Economics of Information Security (WEIS)*, 2015.

Statista, “Digital advertising spending worldwide from 2014 to 2016”, 2016. En ligne : `http://www.statista.com/statistics/246567/global-online-advertising-revenue/`

Statistica, “Number of internet users worldwide from 2000 to 2015”, 2016. En ligne : `http://www.statista.com/statistics/273018/number-of-internet-users-worldwide/`

M. Stockley, “Zeroaccess click fraud botnet coughs back to life”, 2015. En ligne : https://nakedsecurity.sophos.com/2015/01/31/zeroaccess-click-fraud-botnet-coughs-back-to-life/

B. Stone-Gross, R. Stevens, A. Zarras, R. Kemmerer, C. Kruegel, et G. Vigna, “Understanding fraudulent activities in online ad exchanges”, dans *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM, 2011, pp. 279–294.

Support Google, “Publicité utilisant le ciblage par centre d’intérêt.” 2016. En ligne : https://support.google.com/adsense/answer/140381?hl=fr

The Nilson Report, “Global card fraud losses reach $16.31 billion - will exceed $35 billion in 2020 according to the nilson report”, 2015. En ligne : http://www.businesswire.com/news/home/20150804007054/en/Global-Card-Fraud-Losses-Reach-16.31-Billion

K. Thomas, E. Bursztein, C. Grier, G. Ho, N. Jagpal, A. Kapravelos, D. McCoy, A. Nappa, V. Paxson, P. Pearce, N. Provos, et M. A. Rajab, “Ad injection at scale : Assessing deceptive advertisement modifications”, dans *Proceedings of the IEEE Symposium on Security and Privacy*, 2015.

Trustworthy Accountability Group, “Home”, 2016. En ligne : https://www.tagtoday.net/

Virustotal, “c.feed-xml.com domain information”, 2016. En ligne : https://www.virustotal.com/en/domain/c.feed-xml.com/information/

——, 2015. En ligne : https://www.virustotal.com

D. Wang, S. Savage, et G. M. Voelker, “Juice : A longitudinal study of an seo campaign”, dans *Proceedings of the Network and Distributed System (NDSS) Symposium*, 2013.

White House, 2015. En ligne : https://www.whitehouse.gov/net-neutrality

White Ops and Association of National Advertisers, “The bot baseline : Fraud in digital advertising”.

G. Wills, *Symposium on Graph Drawing GD’97*. Springer-Verlag Berlin Heidelberg, 1997.

Wordstream, “Average cost per click around the world”, July 2015. En ligne : http://www.wordstream.com/blog/ws/2015/07/06/average-cost-per-click

G. Xie, M. Iliofotou, T. Karagiannis, M. Faloutsos, et Y. Jin, “Resurf : Reconstructing web-surfing activity from network traffic”, dans *IFIP Networking Conference, 2013.* IEEE, 2013, pp. 1–9.

Q. Zhang, T. Ristenpart, S. Savage, et G. M. Voelker, “Got traffic ? : An evaluation of click traffic providers”, dans *Proceedings of the 2011 Joint WICOW/AIRWeb Workshop on Web Quality*, série WebQuality ’11. New York, NY, USA : ACM, 2011, pp. 19–26.